RG-RSR20-X-28 Series Router

RGOS Configuration Guide, Release 10.4(3b75)

Copyright Statement

Exemption Statement

This document is provided “as is”. The contents of this document are subject to change without any notice. Please obtain the latest information through the Ruijie Networks website. Ruijie Networks endeavors to ensure content accuracy and will not shoulder any responsibility for losses and damages caused due to content omissions, inaccuracies or errors.

Preface

Thank you for using our products. This manual matches the RGOS Release 10.4(3b75).

Audience

This manual is intended for:

- Network engineers
- Technical support and servicing engineers
- Network administrators

Obtaining Technical Assistance

- Ruijie Networks Website: http://www.ruijienetworks.com/
- Service Email: service_rj@ruijienetworks.com
- Technical Support: http://www.ruijienetworks.com/service.aspx
- Technical Support Hotline: +86-4008-111-000

Related Documents

Documents	Description
Command Reference	Describes the related configuration commands, including command modes, parameter descriptions, usage guides, and related examples.
Hardware Installation and Reference Guide	Describes the functional and physical features and provides the device installation steps, hardware troubleshooting, module technical specifications, and specifications and usage guidelines for cables and connectors.

Conventions

This manual uses the following conventions:

Convention	Description
boldface font	Commands, command options, and keywords are in **boldface**.
italic font	Arguments for which you supply values are in *italics*.
[ ]	Elements in square brackets are optional.
{ x | y | z }	Alternative keywords are grouped in braces and separated by vertical bars.
[x | y | z]	Optional alternative keywords are grouped in brackets and separated by

	vertical bars.

Symbols

Note Means reader take note. Notes contain helpful suggestions or references.

Caution Means reader be careful. In this situation, you might do something that could result in equipment damage or loss of data.

Basic Configuration

Configuring the Command Line Interface

This chapter describes how to configurethe command line interface(CLI) to manage network devices.

Command Mode

The management interface of Ruijie network devices has multiple modes and they determine the commands you can use.

To listusable commands in each mode, enter a question mark (?) at the command prompt.

After setting up a session connection to the network device management interface, you enteruser EXEC mode first. In the user EXEC mode, only a few commands are usable with limited functions, for example, the **show**command. The command results are also not saved.

To use all commands, enter privileged EXEC mode with the privileged password. Then you can use all privileged commands and enter global configuration mode.

Using commands in configuration (for example, global configuration or interface configuration) modewill influence the current configuration. If you have saved the configuration information, these commands will be saved and executed when the system restarts. To enter any of the configuration modes, enter global configuration mode in the first.

The following table describescommand modes, access methods, prompts, and exit methods.Suppose the device is named "Ruijie" by default.

The following table summariesmain command modes.

Command mode	Access method	Prompt	Exit or enter the next mode	Remark
User EXEC	Log in.	Ruijie>	To quit this mode, enter the **exit** command. To enter privileged EXEC mode, enter the**enable**command.	Performs basic tests and showssystem information.
Privileged EXEC	In the user EXEC mode, enter the**enable**command.	Ruijie#	To return touser EXEC mode, enter the**disable**command. To enter global configuration mode, enter command **configure**.	Verifiessettings. This mode is password-protected.
Global configuration	Inprivileged EXEC mode, enter the**configure terminal**command.	Ruijie(config)#	To return to privileged EXEC mode, enter command **end** or **exit** or press Ctrl+C. To access the interface configuration mode, enter command **interface**with an interface specified. To access VLAN configuration mode, enterthe **vlan***vlan_id*command.	Executes commands to configure global parameters influencing the entireswitch.

Command mode	Access method	Prompt	Exit or enter the next mode	Remark
Interface configuration	Inglobal configuration mode, enter the **interface**command.	Ruijie(config-if)#	To return to privileged EXEC mode, enter command **end** orpress Ctrl+C. To return to global configuration mode, enter the**exit**command. Moreover, you need specify an interface in the **interface** command.	Configures various interfaces of the device in this mode.
Config-vlan (Vlan Mode)	Inglobal configuration mode, enter the**vlan** vlan-idcommand.	Ruijie(config-vlan)#	To return to privileged EXEC mode, enter command **end** or pressCtrl+C. To return to global configuration mode, enter the**exit**command.	Configures VLAN parameters in this mode.

Obtaining Help

Enter a question mark(?) at the command prompt to obtain a list of commands that are available for each command mode. You can also obtain a list of command keywords beginning with the same character or parameters of each command. See the following table.

Command	Description
Help	Obtainsthe brief description of the help system under any command mode.

abbreviated-command-entry?	Obtains a list of commands that begin with a particular character.
	string.
	For example:
	`Ruijie# di?`
	`dir disable`

abbreviated-command-entry <Tab>	Completes a partial command name.
	For example:
	`Ruijie# show conf<Tab>`
	`Ruijie# show configuration`

?	Listskeywordsassociated witha command. For example: `Ruijie# show ?`
command keyword ?	Listsarguments associated witha command. For example: `Ruijie(config)# snmp-server community ?` WORD SNMP community string

Abbreviating Commands

To abbreviate a command, simply enter part of the command that can uniquely identify the command.

For example, **show configuration**can be abbreviated as follows:

```
Ruijie# show conf
```

Using the No Form and the Default Form

Most commands have the **no**formthatdisables a feature or function, or performs a reversed action of acommand. For example, the **no shutdown**command turns on aninterface, which is the reversed action of the **shutdown** command. You can use the commands without the keyword**no**to enable the features that have been disabled or are disabled by default.

Most configuration commands have the **default**formthat restoresa command setting to its default. The **default** form is disabled for most commands by default. In this case, the **default** and **no**formsgenerally serve the same purpose. However, the default form is enabled for some commandsby default. In this case, the **default** and **no**formsserve different purposes, where the **default**formenables the command and restores the arguments to the default settings.

Understanding CLI Error Messages

The following table describes the error messages that may occur when you use the CLI to manage devices.

Error Message	Meaning	How to Obtain Help
% Ambiguous command: "show c"	The switch cannot identify the unique command because you have entered insufficient characters.	Re-entersthe command with the question mark (?)next tothe ambiguous word. The possible keywords will be listed.
% Incomplete command.	You have not entered the required keywords or arguments.	Re-entersthe command withthespace followed by thequestion mark (?). The possible keywords or arguments will be displayed.
% Invalid input detected at '^' marker.	The symbol (^)indicates the positions of wrong words when user entersa wrong command.	Entersthequestion mark (?) at the command prompt to show the allowed keywords of the command.

Using Historical Commands

The system records the commands you have entered, whichis very useful when you entera long and complex command again.

To re-execute the commands you have entered, perform the following operations.

Operation	Result
Ctrl-P or **Up**	Allows you to browse the preceding command in the historical command records.You can inquire earlier records by repeating this operationfrom the latest record,.
Ctrl-N or **Down**	Allows you to return to a more recent command in the historical command records. You can inquire laterrecords by repeating this operation.

Caution Arrow keys are supported by standard terminals like VT100 series.

Using Editing Features

Editing Shortcut Keys

The following table describes the editing shortcut keys.

Function	Shortcut Key	Description
Move cursor in an editing line	Left direction key or Ctrl+B	Moves the cursor to left by one character.
	Right direction key or Ctrl+F	Moves the cursor to right by one character.
	Ctrl+A	Moves the cursor to the beginning of the command line.
	Ctrl+E	Moves the cursor to the end of the command line.
Delete the entered characters	Backspace	Deletes the character to the left of the cursor.
	Delete	Deletes the character to the right of the cursor.
Scroll up by one line or one page	Return	Scrolls up one line of the display contents anddisplaythe next line. This is used only before the end of the output.
	Space	Scrolls up one page of the displayed contents anddisplaythe next page appear. This is used only before the end of the output.

Sliding Window of Command Lines

You can use this function to edit acommand that exceeds the width of one line. When the editing cursor closes to the right border, the whole command line will move to the left by 20 characters. In this case, the cursor can still be moved back to the previous character or the beginning of the command line.

The following table describes shortcut keys used in this function:

Function	Shortcut key
Moves the cursor to the left by one character.	Left direction key or Ctrl+B
Moves the cursor to the head of a line.	Ctrl+A
Moves the cursor to the right by one character.	Right direction key or Ctrl+F
Moves the cursor to the end of a line.	Ctrl+E

For example, the contents of the **mac-address-table static** command may exceed the screen width. When the cursor approaches the line end for the first time, the whole line move left by 20 characters, and the hidden beginning part is replaced by the symbol ($) on the screen. The line moves left by 20 characters when the cursor reaches the right border.

access-list 199 permit ip host 192.168.180.220 host

$ost 192.168.180.220 host 202.101.99.12

$0.220 host 202.101.99.12 time-range tr

Now you can press **Ctrl+A** to return to the beginning of the command line. In this case, the hidden ending part is replaced by the symbol ($).

access-list 199 permit ip host 192.168.180.220 host 202.101.99.$

Caution The default line width on the terminal is 80 characters.

Combined with historical commands,the sliding window enables you to invoke complicated commands repeatedly. For details about shortcut keys, see the description about the section"Editing Shortcut Keys".

Filtering and Searching CLI Output Information

Filtering and Searching the Output Information of the show Command

Use the following command to search the specified content in theoutput informationof the **show** command.

Command	Description
Ruijie# **show** any-command |**begin**regular-expression	Searches the specified content in the outputinformation of the **show** command andexportsthe first line that contains the specified content and allinformation after the line.

Caution

- You can execute show command in any mode.
- The information to be searched is case sensitive, and the feature is also effective in the following.

Use the following commands to filter the specified content in the output information ofthe **show** command:

Command	Description
Ruijie# **show** any-command **|exclude**regular-expression	Filters the content inthe output information ofthe **show** command andexportsother information excluding the line that includes the specified content.
Ruijie#**show**any-command **|include**regular-expression	Filters the content inthe output information ofthe **show** command and exports the line that includes the specified content.Other information will be filtered.

Caution To search and filter the contents exportedby the **show** command, you must enter the pipeline sign, which is the vertical bar (|) followed by searchand filterrules and contents (characters or strings). The contents are case sensitive.

Using Command Aliases

The system provides the command alias function.You can specify any word as the alias of a command. For example, you can define the word “mygateway” as the alias of the **ip route 0.0.0.0 0.0.0.0 192.1.1.1** command.The effect of enteringthis word is equal to that of enteringthe entire command.

You can use one word to replace one command by configuring an alias for the command. For example, you can define an alias to represent the firstpart of acommand, and then continue to enter the rest parts.

The command that an alias represents must run under the mode you have defined in the current system. In global configuration mode, you can enter **alias?** to list all command modes that can configure aliases.

```
Ruijie(config)#alias ?
  aaa-gs              AAA server group mode
acl                  acl configure mode
  bgp                 Configure bgp Protocol
config              globle configure mode
......
```

An alias supports help information. An alias appears withtheasterisk (*) before it in the following format:

*command-alias=original-command

For example, in EXEC mode, the alias “s” indicates the **show** command by default. You can enter “s?” to obtain the help information on the command and the aliases beginning with ‘s’.

```
Ruijie#s?
*s=show  show  start-chat  start-terminal-service
```

If the command that an alias represents containsmore than one word, the command will be included by the quotation marks. As shown in the following example, you can configure the alias “sv” to replace the **show version**command in EXEC mode.

```
Ruijie#s?
*s=show  *sv=”show version” show  start-chat
start-terminal-service
```

An alias must begin with the first character of the command line entered without any spacebefore it. As shown in the precedingexample, the alias is invalid if you have entereda spacebefore the command.

```
Ruijie# s?
show  start-chat  start-terminal-service
```

An alias can also be used to obtain the help information of command parameters. For example, the alias “ia" represents the **ip address** commandin interface configuration mode.

```
Ruijie(config-if)#ia ?
  A.B.C.D  IP address
  dhcp    IP Address via DHCP
Ruijie(config-if)#ip address
```

The preceding informationlists the parameter information after the command **ip address**, and replaces the alias with the actual command.

A completealias must be entered for use. Otherwise, it can not be identified.

To view the setting of aliases in the system, use the **show aliases** command.

Accessing the CLI

Before using the CLI, you need to use a terminal or PC to connect toanetwork device. Power on the network device.After initializing the hardware and software, you can use the CLI. If the network device is used for the first time, you can only connect to the network deviceoverthe serial port (Console), which is referred to as out-band management. In addition, you can connect and manage the network device through the virtual terminal of Telnet. In either case, you can access the CLI.

Configuring LINE Mode

Configuring LINE Mode

Entering the LINE mode

After entering the specific LINE mode, you can configure the specified line. Use the following command to enter the specified LINE mode:

Command	Function
Ruijie(config)# **line [aux | console | tty | vty**] *first-line* [*last-line*]	Enters the specified LINE mode.

Increasing/Decreasing LINE VTY

The number of line vty is 5 by default. Use the following commands to increase or decrease line vty. 36 line VTYs are supported at most.

Command	Function
Ruijie(config)# **line vty** *line-number*	Increases the number of LINE VTY to the specified value.
Ruijie(config)# **no line vty** *line-number*	Decreases the number of LINE VTY to the specified value.

Configuring the Protocols to Communicate on the Line

Use this command to limit the communication protocol type supported on the line. By default, VTY supports communication of all protocols while TTY does not support the communication of any protocol.

Command	Description
Ruijie# **configure terminal**	Enters the configuration mode.
Ruijie(config)# **line vty *line number***	Enters the line configuration mode.
Ruijie(config-line)# **transport input {all | ssh | telnet | none**}	Configures the protocol to communicate on the line.
Ruijie(config-line)# **no transport input**	Disables the communication of any protocol on the line.
Ruijie(config-line)# **default transport input**	Restores the default value.

Configuring the Access Control List on the Line

Use this command to configure the access control list on the line. No access control list is configured on the line by default.That is, all incoming and outgoing connections are permitted.

Command	Description
Ruijie# **configure terminal**	Enters the configuration mode.
Ruijie(config)# **line vty** *line number*	Enters the line configuration mode.

Command	**Description**
Ruijie(config-line)# **access-class** *access-list-number* **{in | out}**	Configures the access control list on the line.
Ruijie(config-line)# **no access-class** *access-list-number* **{in | out}**	Removes the configuration.

System Upgrade and Maintenance

Overview

Upgrade and maintenance refers to upgrade the main program or CTRL program or upload and download files on the CLI . There are two ways to upgrade programs: use TFTP through a network interface or use Xmodem protocol through a serial interface.

Upgrade and Maintenance Method

Transferring Files by TFTP

There are two ways to transfer files by TFTP: download files from the host to the equipment, or upload files from the equipment to the host.

In the CLI command mode, download the files by performing the following steps:

Before download, first run the TFTP server software on the local host. Then, select the directory of the file to download. Finally, log in to the equipment. In the privileged mode, download the files by using the following commands. If no location is specified, you need to separately input the IP address of the TFTP server.

Command	Function
Ruijie# **copy tftp:** *//location/ filename* **flash:** *filename* [**vrf** *vrfname*]	Download the specified file from the URL on the host to the equipment.

In the CLI command mode, upload the files by performing the following steps:

Before upload, first run the TFTP server software on the local host. Then, select the destination directory for the file to upload at the host. Finally, upload the files by using the following commands in the privileged mode.

Command	Function
Ruijie# **copy flash:** *filename* **tftp:** *//location/filename* [**vrf** *vrfname*]	Upload the specified file from the equipment to the directory specified by the URL on the host. You can also rename the file.

Note

It is necessary to put the tftp link in quotes if the filename of the source file has space. For example:

Note

copy tftp:"//localtion/filename" **flash:**filename [**vrf** vrfname]

Note It is necessary to put the filename in quotes if the filename of the destination file has space. For example:

Note **copy tftp:**//localtion/filename **flash:**"filename" [**vrf** vrfname]

Transferring Files by TFTP IPv6

There are two ways to transfer files by TFTP: download files from the host to the equipment, or upload files from the equipment to the host.

In the CLI command mode, download the files by performing the following steps:

Before download, first run the TFTP server software on the local host. Finally, log in to the equipment. In the privileged mode, download the files by using the following commands.

Command	Function
Ruijie# **copy tftp:** *//location/ filename* **flash:** *filename*	Download the specified file from the URL on the host to the equipment.

In the CLI command mode, upload the files by performing the following steps:

Before upload, first run the TFTP server software on the local host. Then, select the destination directory for the file to upload at the host. Finally, upload the files by using the following commands in the privileged mode.

Command	Function
Ruijie# **copy flash:** *filename* **tftp:** *//location/filename*	Upload the specified file from the equipment to the directory specified by the URL on the host. You can also rename the file.

Caution If location is the local link address, use the following command to specify the egress:

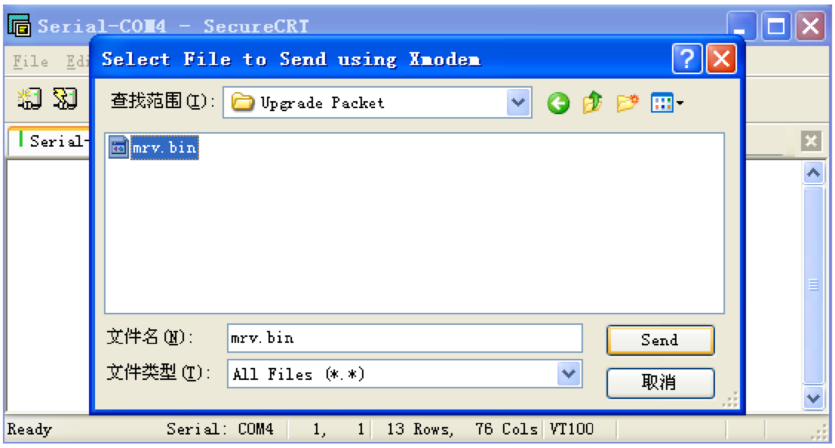
```
Ruijie#copy tftp: flash:
Address of remote host []?fe80::5efe:192.168.195.90
Output Interface: loopback 0
Source filename []?rgos.bin
Extended commands [n]:
Destination filename [rgos.bin]?
```

Transferring Files by XMODEM

There are two ways to transfer files by Xmodem: download files from the host to the equipment, or upload files from the equipment to the host.

In the CLI command mode, download the files by performing the following steps:

Prior to download, first log in to the out-band management interface of the device by using the Windows HyperTerminal. Then, download the files by using the following command in the privileged mode. Finally, select the “Send File” from the “Transfer” menu on the Windows HyperTerminal on the local host, as shown in the following figure:



查找范围(I): Upgrade Packet

文件名(N): mrv.bin

文件类型(T): All Files (*.*)

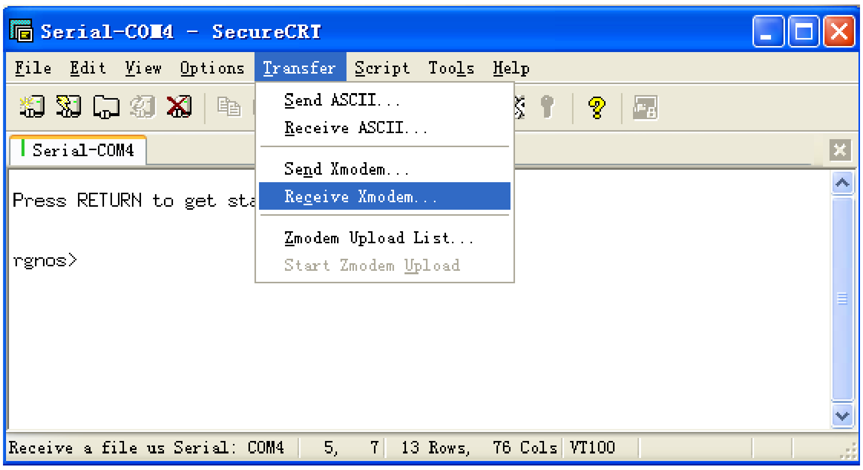
Ready Serial: COM4 1, 1 13 Rows, 76 Cols VT100

In the pop-up dialog box, select the file to download from the File Name field and Xmodem from the Protocol field. Click “Send”, and the Windows HyperTerminal will show the transmission process and packets.

Command	Function
Ruijie# **copy xmodem flash**:*filename*	Download the file from the host to the equipment and name it *filename*.

In the CLI command mode, upload the files by performing the following steps:

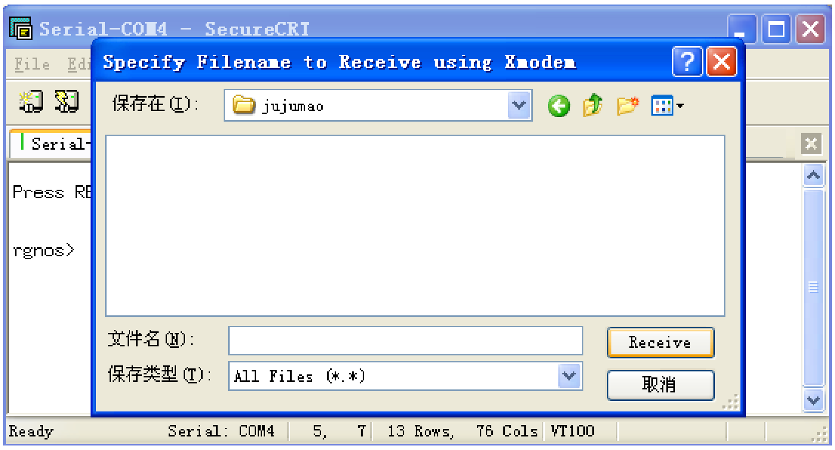
Prior to upload, first log in to the out-band management interface of the switch by using the Windows HyperTerminal. Then, upload the files by using the following command in the privileged mode. Finally, select the “Receive File” from the “Transfer” menu on the Windows HyperTerminal on the local host. It’s shown in the following figure:



```
Press RETURN to get sta

rgnos>
```

Receive a file us Serial: COM4 5, 7 13 Rows, 76 Cols VT100

In the pop-up dialog box, select the storage location for the file to upload and select the “Xmodem” as the reception protocol. Click “Receive”, and the Windows HyperTerminal will further prompt the name of the locally stored file. Click “OK” to start reception. The operation is shown below:


Serial-COM4 - SecureCRT

Specify Filename to Receive using Xmodem

保存在(I): jujumao

Press RE

rgnos>

文件名(N):

保存类型(T): All Files (*.*)

Ready Serial: COM4 5, 7 13 Rows, 76 Cols VT100

Command	Function
Ruijie# **copy flash**:*filename* **xmodem**	Upload the file from the equipment to the host.

Caution It is necessary to put the filename with space in quotes. For example:

```
copy xmodem flash:”filename” OR copy flash:”filename” xmodem
```

Upgrading System

You can transfer the upgrading file to a device through TFTP or Xmodem, no matter the device is box-mount or chassis-mount. After transmission, restart the device. The upgrading file will automatically check and upgrade the system without manual interference.

The upgrade procedure on the box-mount equipment is slightly different from that on the chassis-mount equipment:

On the box-mount equipment, the upgrading file upgrades only its single supervisor engine. After upgrading, the system automatically resets. The equipment works normally after restart.

The chassis-mount equipment includes supervisor engines, line cards and multi-service cards. To upgrade the whole system with a upgrading file, first upgrade the supervisor engine. The system resets. When the equipment restarts, the automatic version synchronization function runs to upgrade line cards and multi-service cards.

Automatic Upgrade: a function running on the supervisor engine that verifies the version consistency for the slave supervisor engine, line cards and multi-service cards. When it is found that the version is not consistent with the one in the master supervisor engine, the function sends the upgrading files to those blades for upgrading so as to keep the version consistence in the whole system.

Caution Whenever you upgrade the master supervisor engine, the slave one (if any) is upgraded at the same time to keep the version consistent. The upgrade of a line card will upgrade all the line cards inserted into the device. Do not power off the device before the upgrade is complete. Otherwise, the upgrade program may be lost.

Caution

Before the chassis-mount device is upgraded, you can check whether the software version of all line cards and supervisor engines are consistent with the upgraded object version by the **show version** command. However, you cannot carry out master-slave switch (such as **redundancy force-switchover**). Otherwise, it will cause the upgrade failure and return to the original version.

Upgrade the chassis-mounted device by the upgrade file:

Confirm the filename of the upgrade file to be loaded is rgos.bin.

Download the file to the device by using the copy command.

If there is a slave supervisor engine on the device, you need to first upgrade the main programs of the master and slave supervisor engines successfully. After upgrading the main program successfully, the system prompts:

```
Upgrade Slave CM MAIN successful!!
Upgrade CM MAIN successful!!
```

Reset the equipment.

After reset, the upgrade file will run automatically. The system prompts:

```
Installing is in process ......
Do not restart your machine before finish !!!!!!
......
```

After the upgrade operation is completed, the system prompts:

```
Installing process finished ......
Restart machine operation is permitted now !!!!!!
```

After the operation of the upgrade file is completed, the system resets automatically and prompts:

```
System restarting, for reason 'Upgrade product !'.
```

After reset, the upgrading operation of the supervisor engines is completed. The system will load and operate the upgrade pack of boards. Moreover, it prompts information in Steps 5 to 6. Instead of the information in Step 7, it prompts:

```
System load main program from install package ......
```

Load the main program of the supervisor engine to operate from the upgrade file directly.

After the main program operates normally, the automatic upgrade function starts. If there is the slave supervisor engine or other modules in the chassis-mount device, the system prompts:

```
A new card is found in slot [1].
System is doing version synchronization checking ......
Current software version in slot [1] is synchronous.
System needn't to do version synchronization for this card ......
```

Or, the system prompts:

```
System is doing version synchronization checking ......
Card in slot [3] need to do version synchronization ......
```

Other Printing Information

```
Version synchronization begain ......
Keep power on, don't draw out the card and don't restart your machine before finished !!!!!!
```

Other Printing Information

```
Transmission is OK, now, card in slot [3] need restart ...
Software installation of card in slot [3] is in process ......
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Software installation of card in slot [3] has finished successfully ......
The version synchronization of card in slot [3] get finished successfully.
```

The former indicates the version of the line card is synchronous and it is not necessary to upgrade again. The latter indicates the version of the line card, and it is necessary to upgrade the line card.

The system will carry out above operation for the slave supervisor engine and each module in turn.

After checking the version consistency on all modules and upgrading, the system will work normally

Caution

During the upgrade or automatic upgrade, the system may prompt that the reboot is not allowed. In this case, neither power off or reset the system nor plug or unplug other modules

Note

Automatic upgrading and checking also applies to the system with hot-plugging modules.

Upgrade the box-mount device by the upgrade file

To upgrade the box-mount device, do Steps 1 to 7, and then the system resets. After that, the equipment runs well.

Configuring Basic Management Features

Overview

Note

For more information about the CLI commands mentioned in this chapter, see the Device Management Command Reference.

Access Control through Command Authorization

Overview

A terminal's network access can be simply managed by using passwords and assigning privileged levels. Passwords restrict access to a netwdydyork or network device. Privileged levels define the commands you can use after logging in to a network device.

For security sake, passwords are stored in a configuration file. Passwords must be kept secure when the configuration file is transmitted, for example, over TFTP or across a network. Passwords are encrypted before they are saved into the configuration file. Plain text passwords becomes cipher text passwords. The **enable secret** command builds on a private encryption algorithm.

Configuring Default Passwords and Privileged Levels

No password at any level is available by default. The default privileged level is 15.

Configuring or Changing the Passwords at Different Levels

Our products provide the following commands for you to configure or change passwords at different levels.

Command	Function
Ruijie(config)# **enable password** [**level** *level*] {*password* | *encryption-type encrypted-password*}	Sets a static password. You can only set a level-15 password when no level-15 security password is configured. If you have set a non-level-15 password, the system will show a message and automatically convert it into a security password. If you have set the same level-15 static password as the level 15 security password, the system will show a warning message. If the encryption type is set to 0, the following password is configured in plain text. If the encryption type is set to 7, the following password is configured in cipher text.
Ruijie(config)# **enable secret** [**level** *level*] {*encryption-type encrypted-password*}	Sets the security password, which provides the same function but a better encryption algorithm than a static password. For security sake, it is recommended you use a security password.

Command	Function
Ruijie(config)#**service password-encryption**	Determines whether to encrypt the related password.
Ruijie# **enable** [*level*], and Ruijie# **disable** [*level*]	Switches between user levels. To move from a lower to a higher level, input the password for the higher level.

When you set a password, the keyword "level" is used to define the password for a specified privileged level. After setting, it only works for the users at that level.

Configuring Multiple Privileged Levels

By default, the system provides only two password-protected levels: normal user (level 1) and privileged user (level 15). You can configure up to 16 hierarchical command levels for each mode. By configuring different passwords at different levels, you can use different sets of commands for different levels.

When no password is set for the privileged user level, you can enter the privileged mode without password authentication. For security, it is recommended you set the password for the privileged user level.

Configuring Command Authorization

To expand the application scope of a command, you can assign it to users at lower levels. On the contrary, to narrow the scope, you can assign it to users at higher levels.

You can use the following commands to authorize users to use a command:

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **service display command privilege**	Enables command-level display. After this function is enabled, you can enter ? to view the level of a command key.
Ruijie(config)# **privilege** *mode* [**all**] {**level** *level* | **reset**} *command-string*	Sets a privileged level for a command. *mode* – The CLI command mode in which you authorize the command. For example, config indicates global configuration mode, exec indicates privileged command mode, and interface indicates interface configuration mode. **all** – Changes the privileges of all the sub-commands of a specified command to the same level. **level** *level* – Authorization level in the range from 0 to 15. **reset**：Restores the command privilege to the default level. *command-string:* Indicates the command you want to authorize.

To restore the configuration for a specified command, use the **no privilege** *mode* [**all**] **level** *level* command in global configuration mode.

Example of Command Authorization Configuration

The following example shows the configuration process that sets the **reload** command and all its sub-commands to level 1, and activates level 1 (by setting the command as “**test**”):

```
Ruijie# configure terminal
Ruijie(config)# service display command privielge
Ruijie(config)# privilege exec all level 1 reload
Ruijie(config)# enable secret level 1 0 test
Ruijie(config)# end
```

Enter level 1, and you can see the command and its subcommands:

```
Ruijie# disable 1
Ruijie> reload ?
  at                   reload at a specific time/date (privilege: 14)
  cancel                cancel pending reload scheme (privilege: 14)
  in                   reload after a time interval (privilege: 14)
  <cr>
```

The following example shows the configuration process that restores the privilege settings of the reload command and all its sub-commands to the default value:

```
Ruijie# configure terminal
Ruijie(config)# privilege exec all reset reload
Ruijie(config)# end
```

Enter the level 1, the privilege setting for the command is removed.

```
Ruijie# disable 1
Ruijie> reload ?
% Unrecognized command.
```

Configuring Line Password Protection

Our products offer password authentication for remote logins (such as Telnet). A password is required for protection. Execute the following command in line configuration mode:

Command	Function
Ruijie(config-line)# **password** [**0** | **7**] *line*	Specifies a line password. 0: The password is configured in plaintext. 7: The password is encrypted by a Ruijie device. Line: the character string of the password to be configured.
Ruijie(config-line)# **secret** { [**0**] *password* | **5** *encrypted-secret* }	Specifies the line's password encrypted by irreversible MD5 **0**: (Optional) specifies the plaintext password text and encrypts it with irreversible MD5 after configuration. *Password*: The password plaintext.. ***5*** *encrypted-secret*: Specifies the password text encrypted by irreversible MD5 and saves it as the encrypted password after configuration.
Ruijie(config-line)# **login**	Enables line password protection.

Note

If no login authentication is configured, password authentication on the line layer will be ignored even when a line password is configured. Login authentication will be discussed in the next section.

Note

If no login authentication is configured, password authentication on the line layer will be ignored even when a line password is configured. Login authentication will be discussed in the next section.

Supporting Session Locking

Our products allow you to lock the session terminal temporarily using the lock command, so as to prevent unauthorized access. To do so, enable the terminal locking function in the line configuration mode, and lock the terminal using the lock command in terminal EXEC mode: The system prompts you for a password for unlocking when you enter any character on a locked terminal. The terminal is locked when your password is authenticated.

Command	Function
Ruijie(config-line)# **lockable**	Enables the function of locking the line terminal.
Ruijie# lock	Locks the current line terminal.

Login Authentication Control

Overview

The previous section discusses how to control access to network devices by configuring a locally stored password. In addition to line password protection and local authentication, in AAA mode, we can authenticate users' management privilege based on usernames and passwords on some servers when they log in to the switch. Take an RADIUS server for example.

With an RADIUS server, the network device sends encrypted user information to the RADIUS server for authentication instead of authenticating them with locally stored credentials. The RADIUS server configures user information consistently like user name, password, shared key, and access policy to facilitate user access management and control and enhance the security of user information.

Configuring Local Users

Our products support identity authentication system based on a local database for local authentication of the method list in AAA mode and local authentication of line login management in non-AAA mode.

To enable username identity authentication, run the following commands in global configuration mode:

Command	Function
Ruijie(config)# **username** *name* [**password** *password* | *encryption-type encrypted password*]	Enables username identity authentication with encrypted password. Encryption type 0 defines a password in plaintext. Encryption type 7 defines an encrypted password.
Ruijie(config)# **username** *name* **secret** { [**0**] *password* | **5** *encrypted-secret* }	Sets the password encrypted by irreversible MD5 for the local user.
Ruijie(config)# **username** *name* [**privilege** *level*]	Sets the privilege level for the user (optional).

Confining the Simultaneously Online Amount of a Local Username

Ruijie products support local usernames confining the simultaneously online amount. By default, local usernames does not limit the simultaneously online amount.

Run the following commands to limit the simultaneously online amount of a local username:

Command	Function
Ruijie(config)# **username** *name* **online amount** *numbers*	Confines the simultaneously online amount of a local username.
Ruijie(config)# **no username** *name* **online amount**	Cancels the limit on the simultaneously online amount of a local username.

After the simultaneously online amount of a local username is set, the number of clients logging in with the username must be within the specified range. When the number exceeds the limit, the username is not allowed to be used for login.

When the simultaneously online amount of a local username is set to 0, no login is allowed with the username by any client, including console login and remote login through this user.

Confining Username Login Mode

Ruijie products support configuration of local username login mode. Login mode can be one type or several types among aux, console ssh and telnet. By default, when there is no restriction on local username login mode, the local username will not confine user login mode.

Run the following demands in global configuration mode to confine local username login mode:

Command	Function
Ruijie(config)#**username** *name* **login mode** { **aux | console | ssh | telnet** }	Confines local username login mode.
Ruijie(config)#**no username** *name* **login mode** { **aux | console | ssh | telnet** }	Cancels restriction on local username login mode.

This command is used to set local username login mode to one type or several types among aux, ssh and telnet. Only the configured login mode is allowed while the other modes are prevented.

Configuring Line Login Authentication

To enable line login identity authentication, run the following commands in line configuration mode:

Command	Function
Ruijie(config-line)# login local	Sets local authentication for line login in non-AAA mode.
Ruijie(config-line)# login authentication {default | list-name}	Sets AAA authentication for line login in AAA mode. The authentication methods in the AAA method list will be used for authentication, including Radius authentication, local authentication and no authentication.

Note

For more information on how to set AAA mode, configure Radius service, and configure the method list, see the sections relating to AAA configuration.

System Time Configuration

Overview

Every switch has its clock, which indicates date (year, month, day) and time (hour, minute, second) and week. When using a switch for the first time, you must configure the clock manually. Of course, you can adjust the clock when necessary. The clock is used for system login that requires you to record the time of an event.

Setting System Time and Date

You can configure the system time on the network device manually. Once configured, the clock will be running continuously even if the network device is powered off. Therefore, unless you need to modify the time, it is not necessary to reconfigure the time.

However, the configuration does not apply to network devices without hardware clock, as the manual time setting actually configures software clock. When the network devices are powered off, you cannot set the time manually.

Caution

RSR10 do not provide hardware clock.

😐 **Support** S2026G, S2026F and RSR10 do not provide hardware clock.

Command	Function
Ruijie# **clock set** hh:mm:ss month day year	Sets system date and time.

For example, change the system time to10:10:12, 2003-6-20:

```
Ruijie# clock set 10:10:12 6 20 2003                 //Set system time and date.
Ruijie# show clock                //Confirm the modification takes effect.
clock: 2003-6-20 10:10:54
```

Showing System Time and Date

You can show system time and date by using the **show clock** command in privileged mode. The following example shows the format:

```
Ruijie# sh clock    //Show the current system time and date.
clock: 2003-5-20 11:11:34
```

Updating Hardware Clock

Some platforms use hardware clock (calendar) to double as software clock. Since battery enables hardware clock to run continuously, hardware clock still runs even though the device is turned off or restarted.

If hardware clock and software clock are out of sync, software clock prevails. Execute the clock update-calendar command to copy date and time from software clock to hardware clock.

In privileged mode, execute the **clock update-calendar** command for software clock to overwrite hardware clock.

😐 **Support**	S2026G, S2026F and RSR10 do not provide hardware clock.

Command	Function
Ruijie# **clock update-calendar**	Updates hardware clock through software clock.

Execute the following command to copy current date and time from software clock to hardware clock.

```
Ruijie# clock update-calendar
```

Scheduled Restart

Overview

This section describes how to use the **reload** [*modifiers*] command to schedule system restarts at specified time. This feature facilitates your operation in some scenarios (for testing, for example). Modifiers is a set of options provided by the **reload** command to increase the command flexibility. The optional modifiers includes **in**, **at** and **cancel**. See the following examples for details:

- **reload in** *mmm* | *hhh:mm* [*string*]

This command sets up the system to restart at regular intervals in the format of mmm or hhh:mm. string is a help prompt. You can give the scheme a name using the string to indicate its purpose. string is a prompt. For example, to reload the system at intervals of 10 minutes for testing, type r**eload in** *10 test*.

- **reload at** hh:mm month day year [string]

This command sets up the system to restart at a specified future time. The parameter year is optional. The year recorded in the system clock is shown by default if no year is specified. As the time span is limited to 30 days, the current system date generally ranges between January 1 and November 30. Therefore, you do not need to specify the year. However, the restart time you have specified can be sometime next January if the system currently shows December. In this situation, you must specify a year to instruct the system to restart next January rather than this January. The system may fail as the restart time is considered to fall in this January by default. string is used in a similar way. For example, input **reload at** *08:30 11 1 newday* if the current system time is 14:31 on January 10, 2005 and you want the system to reload tomorrow. If the current system time is 14:31 on December 10, 2005 and you want the system to reload at 12:00 a.m. on January 1, 2006, input **reload at** *12:00 1 1 2006 newyear*.

- **reload cancel**

This command deletes a user-defined restart scheme. As mentioned earlier, you have specified the system to reload at 8:30 a.m. tomorrow, the setting will be canceled after you input **reload cancel**.

Note

If the system supports clock function, you can use option at. Before the use, it is recommended you configure the system clock as required. If a restart scheme has been set before, subsequent settings will overwrite previous settings. If you have set a restart scheme and you restart the system before the scheme takes effect, the scheme will be lost.

Note

The span between the time indicated in the restart scheme to the current time must be within the range of 200 days and must be later than the current system time. Besides, after you have set reload, you should not set system clock. Otherwise, your setting may fail if the system time is later than the reload time.

Specifying the System to Restart at the Specified Time

In privileged mode, you can configure system reload at the specified time using the following commands:

Command	Function
Ruijie# **reload at** *hh:mm month day* [*year*] [*reload-reason*]	Reloads at hh:mm,month day,year. reload-reason (if any); indicates the reason that the system reloads.

The following example shows an example of system reload at 12:00 a.m. January 11, 2005 (suppose the current system clock is 8:30 a.m. January 11, 2005):

```
Ruijie# reload at 12:00 1 11 2005 midday   //Set the reload time and date.
Ruijie# show reload //Confirm the modification takes effect.
Reload scheduled for 2005-01-11 12:00  (in 3 hours 29 minutes)16581 seconds.
At 2005-01-11 12:00
Reload reason: midday
```

Specifying the System to Restart after a Period of Time

In privileged mode, you can configure the system reload at the specified time using the following commands:

Command	Function
Ruijie# **reload in** *mmm* [*reload-reason*]	Configures the system reload in mmm minutes, where the reload reason is described in reload-reason (if entered)
Ruijie# **reload in** *hhh:mm* [*reload-reason*]	Configure the system reload in hhh hours and mm minutes, where the reload reason is described in reload-reason (if entered)

The following example shows how to reload the system in 125 minutes (assume that the current system time is 12:00 a.m. January 10, 2005):

```
Ruijie# reload in 125 test //Set the system reload time
Or
Ruijie# reload in 2:5 test     //Set the system reload time
Ruijie# show reload //Confirm whether the restart time change takes effect
System will reload in 7485 seconds.
```

Immediate Restart

The **reload** command without any parameter will restart the device immediately. In privileged mode, you can restart the system immediately by using the **reload** command.

Deleting the Configured Restart Scheme

In privileged mode, use the following command to delete the configured restart scheme:

Command	Function
Ruijie# **reload cancel**	Deletes the configured restart scheme.

If no reload scheme is used, an error message appears.

Configuring a System Name and Prompt

Overview

For easier management, you can configure a system name for the switch to identify it. If you configure a system name that contains more than 32 characters, the first 32 characters are used as the system prompt. The prompt varies with the system name. The system is named Ruijie by default.

Configuring a System Name

Our products provide the following commands to configure a system name in global configuration mode:

Command	Function
Ruijie(Config)# **hostname** *name*	Sets the system name. The name must contain up to 63 printable characters.

To restore the name to the default value, use the **no hostname** command in global configuration mode. The following example changes the device name to RGOS:

```
Ruijie# configure terminal     //Enter global configuration mode.
Ruijie(config)# hostname RGOS      //Set the equipment name to RGOS
RGOS(config)#                  //The name has been modified successfully.
```

Configuring a Command Prompt

The system name appears as the default if you have not configured any command prompt. (If the system name exceeds 32 characters, the first 32 characters will be blocked.) The prompt varies with the system name. You can use the **prompt** command to configure a command prompt in global configuration mode. The command prompt only applies in EXEC mode.

Command	Function
Ruijie# **prompt** *string*	Sets the command prompt with printable characters. If the name exceeds 32 characters, the first 32 characters are blocked.

To restore the prompt to the default value, use the **no prompt** command in global configuration mode.

Banner Configuration

Overview

When a user logs in to the switch, you may need to give the user useful information through a banner. There are two kinds of banners: message-of-the-day (MOTD) and login banner. The MOTD is unique to users who connect with switches. When users log in, the notification message will appear on the terminal. MOTD allows you to send urgent messages (for example, the system is shutting down) to network users. The login banner also appears on all connected terminals. It provides some common login messages. By default, no MOTD and login banners are configured.

Configuring a Message-of-the-Day

You can create a notification of single or multi-line messages that appears when a user logs in the switch. To configure the message of the day, execute the following commands in global configuration mode:

Command	Function
Ruijie(Config)# **banner motd** *c message c*	Specifies the message of the day, with c being the delimiter, for example, a pound sign (&). After entering the delimiter, press Enter. Now, you can type text. You need to input the delimiter and then press Enter. Note that if you type additional characters after the ending delimiter, these characters will be discarded by the system. Also note that you cannot use the delimiter in the message and the message can contain a maximum of 255 bytes.

To delete the MOTD, use the no banner motd command in global configuration mode. The following example describes how to configure a MOTD. The # symbol is used as the delimiter, and the text is "Notice: system will shutdown on July 6th."

```
Ruijie(config)# banner motd #                      //Start delimiter.
Enter TEXT message.  End with the character '#'.
Notice: system will shutdown on July 6th.#             //End delimiter.
Ruijie(config)#
```

Configuring a Login Banner

To configure a login banner, execute the following commands in global configuration mode:

Command	Function
Ruijie(Config)# **banner login** *c message c*	Specifies the text of the login banner, with c being the delimiter, for example, a pound sign (&). After entering the delimiter, press Enter. Now, you can start to type text. You need to input the delimiter and then press Enter. Note that if you type additional characters after the end delimiter, these characters will be discarded by the system. Also note that you cannot use the delimiter in the text of the login banner and the text can contain a maximum of 255 bytes.

To delete the login banner, use the **no banner login** command in global configuration mode.

The following example shows how to configure a login banner. The pound sign (#) is used as the starting and end delimiters and the text of the login banner is "Access for authorized users only. Please enter your password."

```
Ruijie(config)# banner login # //Start delimiter
Enter TEXT message. End with the character '#'.
Access for authorized users only. Please enter your password.
#   //End delimiter
Ruijie(config)#
```

Displaying a Banner

A banner is displayed when you log in the network device. See the following example:

```
C:\>telnet 192.168.65.236
 Notice: system will shutdown on July 6th.
 Access for authorized users only. Please enter your password.
 User Access Verification
Password:
```

“Notice: system will shutdown on July 6th." is a MOTD banner and "Access for authorized users only. Please enter your password." is a login banner.

Viewing System Information

Overview

You can check some system information using the show command on the command-line interface, such as version and device information.

Viewing System Information and Version

System information includes description, power-on time, hardware version, software version, BOOT-layer software version, and CTRL-layer software version.This information helps you know the system better. You can show system information using the following commands in privileged mode.

Command	Function
Ruijie# **show version**	Shows system information.

Note

For a sequence number, run the **show version** command on the main program interface to view SYSTEMUPTIME in the form of DD:HH:MM:SS.

Note

During upgrading, the running software version may differ from the version in the file system. In this case, the main program version shown by running the show version command is the one running in the memory, but the Boot/Ctrl version is the one stored in the flash memory.

Viewing Hardware Entity Information

Hardware information relate to physical devices, slots and modules assembled in a device. The information on a device includes description, number of slots,slot information, slot number, description of the module on the slot (empty description if no module is plugged in the slot), the number of physical ports of the module in the slot, and the maximum number of ports supported in the slot (the number of ports on the plugged module). You may use the following commands to show the information about the device and slots in privileged mode:

Command	Function
Ruijie# **show version devices**	Shows the current device information.
Ruijie# **show version slots**	Shows current information about slots and modules.

Setting Console Rate

Overview

The device provides a console interface for management. When using the switch for the first time, you need to perform configuration through the console interface. You can change the console rate on the device if necessary. Note that the rate of the terminal used to manage the switch must be the same as that of the console interface on the switch.

Setting Console Rate

In line configuration mode, execute the following command to set the console rate:

Command	Function
Ruijie(config-line)# **speed** *speed*	Sets transmission rate in bps on the console interface. For a serial interface, you can only set the transmission rate to any one of the following values: 9600, 19200, 38400, 57600 and 115200 bps, with 9600 bps by default.

This example shows how to configure the baud rate of the serial interface to 57600 bps:

```
Ruijie# configure terminal  //Enter global configuration mode.
Ruijie(config)# line console 0     //Enter the console line configuration mode
Ruijie(config-line)# speed 57600   //Set the console rate to 57600bps
Ruijie(config-line)# end       //Return to the privileged mode
Ruijie# show line console 0    //View the console configuration
CON    Type   speed   Overruns
* 0    CON    57600   0
Line 0, Location: "", Type: "vt100"
Length: 25 lines, Width: 80 columns
Special Chars: Escape  Disconnect  Activation
             ^^x    none       ^M
Timeouts:    Idle EXEC    Idle Session
             never         never
History is enabled, history size is 10.
Total input: 22 bytes
Total output: 115 bytes
```

```
Data overflow: 0 bytes
stop rx interrupt: 0 times
Modem: READY
```

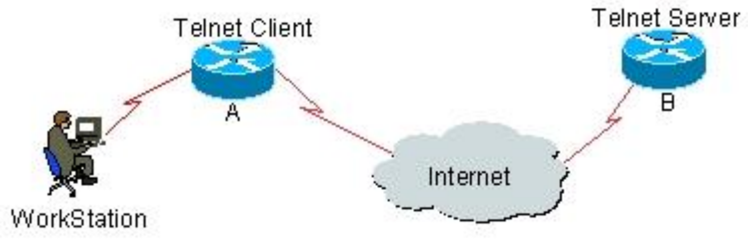
Configuring Telnet

Overview

Telnet, as an application layer protocol in the TCP/IP protocol suite, provides the specifications for remote login and virtual terminal communication. The Telnet Client service is used by a local or remote user who has logged onto the local network device to work with the Telnet Client program to access other remote system resources on the network. As shown below, after setting up a connection with Switch A through the terminal emulation program or Telnet, you can log in the Switch B for management and configuration using the telnet command.

Ruijie's telnet program supports IPV4 and IPV6 addresses. The telnet server can receive IPV4 and IPV6 telnet connection requests. The telnet client can send connection requests to an IPV4 or IPV6 host.

Figure 1



Using Telnet Client

You can log in to a remote device by using the **telnet** command on the switch.

Command	Function
Ruijie# **telnet** *host* [*port*] [/**source** {**ip** *A.B.C.D* **ipv6** *X:X:X::X* | **interface** *interface-name*}] [/**vrf** *vrf-name*]	Logs in to a remote device via Telnet. host may be an IPv4 or IPv6 host name or an IPv4 or IPv6 address. For optional parameters, refer to relevant Telnet command section in Basic Configuration Management Command.

The following example shows how to establish a Telnet session and manage the remote device with the IP address 192.168.65.119:

```
Ruijie# telnet 192.168.65.119  //Establish the telnet session to the remote device
Trying 192.168.65.119 ... Open
User Access Verification       //Enter into the login interface of the remote device
Password:
```

The following example shows how to establish a Telnet session and manage the remote device with the IPv6 address 2AAA:BBBB::CCCC:

```
Ruijie# telnet 2AAA:BBBB::CCCC //Establish the telnet session to the remote device
Trying 2AAA:BBBB::CCCC ... Open
User Access Verification       //Enter into the login interface of the remote device
```

```
Password:
```

Using Telnet Server

Use the following command to enable the Telnet server service for network devices:

Command	Function
Ruijie(config)# **enable service telnet-server**	Enables the Telnet server service. This command enables IPv4 and IPv6 services concurrently.

Setting Connection Timeout

Overview

You can control the connections of a device (including the accepted connections and sessions between the device and a remote terminal) by configuring the connection timeout for the device. When the idle time exceeds the set value and no input or output is found, this connection will be released.

Connection Timeout

When there is no information running through an accepted connection within a specified time, the server will release this connection.

Our products provide commands for you to configure the connection timeout in line configuration mode.

Command	Function
Ruijie(Config-line)#**exec-timeout** *minutes* [*seconds*]	Configures the timeout for the accepted connection. When the configured time is due and there is no input, this connection will be released. *minutes*: timeout in minutes; *seconds*: timeout in seconds.

You can cancel the connection timeout by using the **no exec-timeout** command in line configuration mode.

```
Ruijie# configure terminal  //Enter global configuration mode.
Ruijie# line vty 0          //Enter the line configuration mode
Ruijie(config-line)#exec-timeout 20    //Set the timeout to 20min
```

Session Timeout

When there is no input for the session established with a remote terminal over the current line within the specified time, the session will be released and the remote terminal becomes idle.

RGOS provides commands in line configuration mode to configure the timeout for sessions with a remote terminal.

Command	Function

Command	Function
Ruijie(Config-line)#**session-timeout** *minutes* [**output**]	Configures the timeout for the session set up with the remote terminal over the line. If there is no input within the specified time, this session will be released. *minutes*: timeout in minutes; **output**: Determines whether the session has expired using output data.

You can remove the timeout setting for the session set up with the remote terminal by using the **no exec-timeout** command in the line configuration mode.

```
Ruijie# configure terminal //Enter global configuration mode.
Ruijie(config)# line vty 0         //Enter the line configuration mode
Ruijie(config-line)# session-timeout 20    //Set the session timeout to 20min
```

Executing the Commands for Executable Batch Files

During the process of system management, it is sometimes necessary to enter multiple configuration commands to manage a function. It takes a long time to enter all the commands on CLI, causing errors. To solve this problem, you can include all the commands into a batch file by taking configuration steps. Then, you can execute the batch file for configuration when necessary.

Command	Function
Ruijie# **execute** {[**flash**:] *filename*}	Executes a batch file.

For example, the batch file line_rcms_script.text enables the reversed Telnet function on all the asynchronous interfaces as shown below:

```
configure terminal
line tty 1 16
transport input all
no exec
end
```

Result:

```
Ruijie# execute flash:line_rcms_script.text
executing script file line_rcms_script.text ......
executing done
Ruijie# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# line vty 1 16
Ruijie(config-line)# transport input all
Ruijie(config-line)# no exec
Ruijie(config-line)# end
```

Note

The file name and contents of a batch file can be specified. You can send an edited batch file to the flash memory of the network device in TFTP mode. The contents of the batch file will synchronize the input completely. Hence, it is necessary to edit the contents of the batch file in the sequence that CIL commands are configured. For some interactive commands, it is necessary to write corresponding response information into the batch file to ensure that the commands can be executed normally.

Caution

Files exceeding 128 KB may cause batch processing to fail. For batch processing, split a large file into a number of files, each of which is smaller than 128 KB.

Setting a Service Switch

During operations, you can adjust services dynamically to enable or disable specified services (SNMP Server/SSH Server/Telnet Server/Web Server).

Command	Function
Ruijie(Config)# **enable service snmp-agent**	Enables SNMP Server.
Ruijie(Config)# **enable service ssh-sesrver**	Enables SSH Server.
Ruijie(Config)# **enable service telnet-server**	Enables Telnet Server.
Ruijie(Config)# **enable service web-server**	Enables http and https servers.

In configuration mode, you can use the **no enable service** command to disable corresponding services.

```
Ruijie# configure terminal             //Enter global configuration mode.
Ruijie(config)# enable service ssh-server  //Enable SSH Server
```

To enable http service only, use the following command:

```
Ruijie(config)# enable service web-server http
```

To enable https service only, use the following command:

```
Ruijie(config)# enable service web-server https
```

Note

The **enable service web-server** command can be followed by three optional keywords:

enable service web-server [*http* | *https* | *all*]

If the command is followed by no keyword or by all, the command enables http and https services. Followed by http, the command enables http service only. Followed by https, the command enables https service only. This command and related HTTP service commands discussed later do not necessarily enable you to access the web management page through the browser. These commands only enable the HTTP service and provide an HTTP access channel. To access the web management page, upload a compressed web management work package in upd format to the flash memory of your device. These commands are not designed only for web management. Instead, they support HTTP detection, redirection, and flash file download on the device.

Caution

For the HTTP server to work after the HTTP service is enabled, store the server certificate and private key in the root directory of the file system on the device. Name the server certificate httpd_cert.crt and private key file httpd_key.pem, and upload these files to the root directory through a TFTP server.

Setting HTTP Parameters

When using the integrated Web for management, you can adjust HTTP parameters, and specify service ports or login authentication methods.

Command	Function
Ruijie(Config)# **ip http port** *number*	Specifies HTTP service port, 80 by default.
Ruijie(Config)# **ip http authentication** {**enable** | **local**}	Sets Web login authentication method, which is enable by default. **enable**: Uses the password set by the enable password or enable secret command for authentication, where the password must be 15 levels. **local**: Uses the username and password set by the username command for authentication, where the user must be bound with 15-level access rights.

In configuration mode, you can use the **no** form of the command to restore the setting to the default value. The following example enables the HTTP Server, sets the service port to 8080, and uses the local username for login authentication.

```
Ruijie# configure terminal             //Enter global configuration mode.
Ruijie(config)# enable service web-server http     //Enable http Server
Ruijie(config)# username name password pass //Set local user
Ruijie(config)# username name privilege 15     //Bind user right
Ruijie(config)# ip http port 8080          //Set service port
Ruijie(config)# ip http authentication local //Set authentication method
```

Use the following command to configure an HTTPS service port.

Command	Function
Ruijie(Config)# **ip http secure-port** *number*	Specifies the HTTP service port. (default:443)

In configuration mode, you can use the **no** form of the command to restore the setting to the default value. The following example enables the HTTP Server and sets the service port to 4443.

```
Ruijie# configure terminal             //Enter global configuration mode.
Ruijie(config)# enable service web-server https//Enable https Server
Ruijie(config)# ip http secure-port 4443
```

Use the following command to verify the status of WEB server.

```
Ruijie# show web-server status
http server status : enabled
http server port : 8080
https server status:  enabled
https server port: 4443
```

Caution

Avoid configuring http and https service ports to the same value. If https service is enabled after http service, and the port is configured to the same port as http service, you can only access https service through this port, and http service will be blocked temporarily until https service port is changed or the service is disabled.

Setting Multi-boot Function

Overview

By default, the device searches for the main program file and boots it in the built-in flash memory. If the main program file is damaged due to upgrade failure, formatted flash memory, for example, the device may fail to boot the system.

Some Ruijie products support the multi-boot function, which enables you to boot the device using main program files from local flash memory, a removable disk (USB drive or SD card) or a remote TFTP server. When the device starts, the system boots the main programs by boot pirority in descending order until it boots successfully or all programs are filed. Multi-boot function is mandatory for some environments with higher demands on reliability and availability.

Product support	Only the RSR20, RSR30, R2700 V5.0, RSR50, RSR50E and NPE50 series of routers and the S86 series of switches currently support the multi-boot function. Unless otherwise stated, this section applies to the above products.

This following examples describe how to use multi-boot for redundant backup of the main program.

Configuring the Main Boot Program

You can use the following command to configure the main boot program and specify the boot priority. The system will boot the main program based on priority in descending order with 1 being the highest and 10 being the lowest priority.

Command	Function
Ruijie(Config)#**boot system** *priority* prefix:/ [directory/] filename	Sets the main boot program and specify its priority. The boot priority is in the range of 1 to 10, with 1 being the highest.

Caution

Using URL prefix to locate a file is only supported in 10.4(2) and higher versions. For details, refer to the File System Configuration Guide. Path is used to locate a file in a version lower than 10.4(2), for example, usb0:/backup/rgos.bin represents rgos.bin in the backup directory on the first USB device.

flash:/rgos.bin indicates the rgos.bin file under the Flash root directory.

Caution

Supported URL prefixes vary with platforms. To show the currently supported URL prefixes, run the following command:

```
Ruijie (config) # boot system 2 ?
  flash:   Boot from flash: file system
  tftp:    Boot from tftp server
  usb0:   Boot from usb0: file system
  usb1:   Boot from usb1: file system
```

Caution

Multi-boot is not allowed during ISSU operations (see “Upgrading ISSU”).

By default, the bootable main program is flash:/rgos.bin with the priority of 5.

Caution

Since the system uses this command in the early stage of booting, the configuration is saved in the Boot ROM rather than in the configuration file.

Specifying a File in Local Flash Memory

The following example sets the file on the local flash memory as the main program.

```
Ruijie(config)# boot system 5 flash:/rgos.bin
```

Note

When you specify a local file through prefix, the path following “:” must be an absolute path.

When you configure the boot system command, the system will check the validity of the main program in the local flash memory. You can configure the command successfully only when the main program meets the following requirements.

- The main program must exist.
- The main program is a legal RGOS main program.
- The main program is complete and passes CRC check.

If any requirement is unmet, the systme will dispay an error message, for example:

```
Ruijie(config)# boot system 5 flash:/foo.bin
Set boot system file error:[ flash:/foo.bin] does not exist!
```

In addition, a priority can be set for more than one main program. Otherwise, the system will display an error message and print the current main program list for your selection. For instance:

```
Ruijie(config)# boot system 5 flash:/rgos.bin
Ruijie(config)# boot system 5 flash:/rgos_bak.bin
Set boot system file error: priority 5 has been assigned to file [ flash:/rgos.bin].
Boot system config:
=====================================================
Prio     Size            Modified  Name
---- --------- ------------------- ------------------
  1
  2
  3
  4
  5    3205120 2008-08-26 05:22:46 flash:/rgos.bin
  6
  7
  8
  9
 10
======================================================
Ruijie(config)# boot system 6 flash:/rgos_bak.bin
```

Specifying a File on a Removable Storage Device

The same procedure applies for saving the main program as a file on a removable storage device and in the local flash memory. The only difference is that the system does not check the file for existence or validity when you save it on a removable storage device.

Caution

The file is not checked so that you can configure the device remotely without having to insert a USB drive containing a valid main program into the device. However, you must enter a correct filename.

Do as follows to set up the device to boot from a USB drive:

```
Ruijie(config)# boot system 1 usb1:/rgos.bin
```

Note

Currently, the device cannot start a RGOS installation package earlier than 10.4 (3) from a USB drive.

Specifying a File on a Remote TFTP Server

Do as follows to set up the device to boot from a TFTP server:

```
Ruijie(config)# boot system 2 tftp://192.168.7.24/rgos.bin
```

Note

Currently, the device cannot start a RGOS installation package earlier than 10.4 (3) from a TFTP server.

For the device to download the main program through the TFTP protocol during the boot process, use the boot ip command to configure a correct local IP address used for TFTP address:

Command	Function
Ruijie(config)# **boot ip** *local-ip*	Configures a local IP address for TFTP transfer during the boot process.
Ruijie(config)# **no boot ip**	Clear the boot ip configuration.

Caution

This configuration is stored in the Boot ROM rather than in a configuration file, as the system must use the configuration early in the boot process.

Caution

Ensure that the built-in flash memory contains sufficient free space for the boot file when booting from a TFTP server. During the boot process, the file is hidden in the flash memory. Clear it before the next boot.

Do as follows to configure the IP address for the device to boot up:

```
Ruijie(config)# boot ip 192.168.7.11
```

If no boot ip address is specified, the device cannot load main program files from the TFTP server during the boot process due to communication failure. The following message appears on the screen:

```
Load program file: [tftp://192.168.7.24/rgos.bin]
[Failed] (Boot IP was not assigned)
Load program file: [/rgos.bin]
[OK]
Executing program, launch at: 0x00010000
......
```

Modifying the Boot Priority of Main Program

The **boot system** command can also be used to modify the boot priority of the main program. Aussme that the configured boot main program list is shown as follows:

```
Ruijie# show boot system
Boot system config:
=====================================================
Prio     Size            Modified  Name
---- --------- ------------------- ------------------
  1
  2
  3
  4
  5    3205120 2008-08-26 05:22:46 flash:/rgos.bin
  6
  7
  8    3205120 2008-08-26 05:25:09 flash:/rgos_bak.bin
  9
 10
======================================================
```

To set the boot priority of flash:/rgos_bak.bin to 1, run the following command:

```
Ruijie(config)# boot system 1 flash:/rgos_bak.bin
File [flash:/rgos_bak.bin] has been configured with priority 8,
Change the priority to [1]? [yes] yes
```

The result is as follows:

```
Ruijie# show boot system
Boot system config:
=====================================================
Prio     Size            Modified  Name
---- --------- ------------------- ------------------
  1    3205120 2008-08-26 05:25:09 flash:/rgos_bak.bin
  2
  3
  4
  5    3205120 2008-08-26 05:22:46 flash:/rgos.bin
  6
  7
  8
  9
 10
======================================================
```

Deleting the Main Boot Program

You can use the following command to delete the main boot program.

Command	Function
Ruijie(Config)# **no boot system** [*priority*]	Deletes the main boot program. The boot priority is in the range of 1 to 10. If no priority is specified, all the main boot programs will be reset.

Use the following command to delete the main program with the prioirty of 8. During the process of deletion, the system displays the corresponding main program name and prompts you for confirmation.

```
Ruijie(config)# no boot system 8
Delete boot system config: [Priority: 8; File Name: flash:/rgos_bak.bin]? [no] yes
```

Use the following command to clear all the main boot programs.

```
Ruijie(config)# no boot system
```

```
Clear ALL boot system config? [no] yes
```

Caution

If you have not configured the main boot program after using the **no boot system** command to clear all main boot programs, the system will automatically restore to the default setting during the next booting process (the bootable main program is flash:/rgos.bin with the priority of 5).

Showing the Configuration of Multi-boot

You can use the following command to show the configuration of multi-boot.

Command	Function
Ruijie# **show boot system**	Show the configuration of the main boot program.
Ruijie# **show boot ip**	Shows the local IP address used by the device during the boot process.

The local IP address for booting up the device is shown as follows:

```
Ruijie# show boot ip
System boot ip: [192.168.7.11]
```

Use the following command to show the main program and its boot priority.

```
Ruijie# show boot system
Boot system config:
=====================================================
Prio     Size            Modified  Name
---- --------- ------------------- ------------------
  1
  2
  3
  4
  5    3205120 2008-08-26 05:22:46 flash:/rgos.bin
  6
  7
  8    3205120 2008-08-26 05:25:09 flash:/rgos_bak.bin
  9
 10
======================================================
```

Note

The size and modification time are not shown for files on a remote TFTP server. The size and modification time are shown as N/A for such files.

Note

If the related main program does not exist when you run the **show boot system** command, the size and modification time of the file is also shown as N/A.

Configuration Example

The following example shows how the device boots up:

- The device boots from the USB drive in USB port 1 that contains a legal main program file in its root directory;
- The device boots from rgos.bin in the root directory of the built-in flash memory if no USB drive is available;
- The device will boot from the backup main program file rgos_bak.bin if rgos.bin is damaged or lost;
- The device will download and boot from a main program file from the remote TFTP server if the backup main program file fails possibly because the built-in flash memory is formatted.

Step 1: Configure the default main program.

```
Ruijie(config)# boot system 5 flash:/rgos.bin
```

Since the device is configure with the main program flash;/rgos.bin with priority of 5 during intialization, this step can be skipped.

Generally, it is recommended that you set the priority of an active main program to medium so that you can configure other main programs with a higher or lower priroity.

Step 2: configure the backup main program.

The backup main program should have a priority slightly lower than the active main program.

```
Ruijie(config)# boot system 8 flash:/rgos_bak.bin
```

Step 3: Configure the name of the main program for booting from a remote TFTP server and the boot IP address.

Normally, the device is set up to boot from a remote TFTP server only when its built-in flash memory is damaged. Therefore, boot from TFTP is set to the lowest priority:

```
Ruijie(config)# boot ip 192.168.7.11
Ruijie(config)# boot system 10 tftp://192.168.7.24/rgos.bin
```

Step 4: Configure the name of the main program for USB boot.

Boot from a USB drive applies when a temporary version must be quickly deployed for trial run.

For a device to boot first from a USB drive, insert the USB drive that contains only the temporary software version and restart the device. To clear the temporary version, remove the USB drive and restart the device. The device will automatically boot from the main program in the built-in flash memory. Booting from a USB drive simplifies the deployment of a temporary version and shortens the downtime due to version upgrade.

Do as follows to configure the name of the main program and set up the device to boot first from a USB drive:

```
Ruijie(config)# boot system 1 usb1:/rgos.bin
```

Step 5: Check the main program name and boot priority.

You can run the **show boot** system command to view confiugration.

```
Ruijie# show boot system
Boot system config:
=====================================================
Prio     Size            Modified  Name
---- --------- ------------------- ------------------
  1
  2
  3
  4
  5    3205120 2008-08-26 05:22:46 flash:/rgos.bin
  6
  7
  8    3205120 2008-08-26 05:25:09 flash:/rgos_bak.bin
  9
 10        N/A                 N/A tftp://192.168.7.24/
                                   rgos.bin
======================================================
```

Setting Startup Configuration File

Some Ruijie products can speficy a startup configuration file, which is stored in the flash memory, on a removable storage device (for example, USB drive, SD card) or remote TFTP server.

Once configured, a device can obtain a file from a specified location as the startup configuration file.

Product support	Only the RSR20, RSR30, R2700 V5.0, RSR50, RSR50E and NPE50 series of routers and the S86 series of switches currently support this function. Unless otherwise stated, this section applies to the above products.

This following examples describe how to specify a startup configuration file.

Configuring the Startup Configuration File

You can use the following command to configure the startup configuration file.

Command	Function
Ruijie(Config)#**boot config** *prefix:/ [directory/] filename*	Sets the startup configuration file.
Ruijie(Config)#**no boot config**	Clears the startup configuration file.

Note

You can view the configuration file using the command line help, for example:

```
Ruijie(config)#boot config ?
```

```
 flash:   Startup-config filename
 usb0:   Startup-config filename
 usb1:   Startup-config filename
```

The system loads a configuration file as follows:

- If the **service config** command is absent, configuration files are loaded in the following order: the startup configuration file specified by the **boot config** command, /config.text, the network startup configuration file configured by the **boot network** command, and the default factory configuration (null configuration).
- If the **service config** command is present, configuration files are loaded in the following order: the network startup configuration file configured by the **boot network** command, the startup configuration file specified by the **boot config** command, /config.text, and the default factory configuration (null configuration).
- While loading configuration files in order, the system will not load another configuration files until one configuration file is loaded successfully.

Caution

For the service config and boot config commands, refer to the following examples.
Because the system needs to use the configuration of this command in the early stage of booting, this configuration is stored in Boot ROM rather than in the configuration file.

When using the **write [memory]** command to store the startup configuration file, the system will save it as follows:

- If the **boot config** command is not used, the system stores the configuration in the flash:/config.text file in the built-in flash memory by default.
- If the **boot config** command is used to configure a startup configuration file and the file exists, the system stores the configuration in the startup configuration file.
- If the **boot config** command is used to configure a startup configuration file but the configuration file does not exist, then:
- If the device where the configuration file is located exists, the system will automatically create the specified configuration file and store it into the system configuration.
- If the device where the configuration file is located does not exist (for instance, the start cofniguration file is stored on a removable storage device such as a USB drive or SD card, but the device is not loaded when the system runs the **write [memory]** command), the system will prompt you whether to save the configuration into the default startup configuration file flash:/config.text and act as required.

The following example sets a file on the USB drive as the startup configuration file and demonstrates how to run the write command before and after removing the USB drive.

Set the file on the USB drive as the startup configuration file.

```
Ruijie(config)# boot config usb1:/config.text
```

Run the **write** command before removing the USB drive to save the current configuation into the file specified by the boot config command.

```
Ruijie# write
Building configuration...
```

```
Write to boot config file: [usb1:/config.text]
[OK]
```

Run the **write** command after removing the USB drive. The system will prompt you whether to save the current configuration into the default start configuraiton file /config.

```
Ruijie# usb remove 1
0:1:1:38 Ruijie: USB-5-USB_DISK_REMOVED: USB Device <USB Mass Storage Device> Removed!
Ruijie# write
Building configuration...
Write to boot config file: [usb1:/config.text]
[Failed]
The device [usb1] does not exist, write to the default config file [flash:/config.text]? [no]
yes
Write to the default config file: [flash:/config.text]
[OK]
```

Configuring the Network Start Configuration File

You can use the following command to configure the network startup configuration file.

Command	Function
Ruijie(Config)# **boot network tftp** :// *location* / *filename*	Configures the network startup configuration file.
Ruijie(Config)#**no boot network**	Clears the network startup configuration file.

When the device starts, the system loads the configuration file as follows;

- If the service config command is absent, configuration files are loaded in the following order: the startup configuration file specified by the boot config command, /config.text, the network startup configuration file cofnigured by the boot network command, and the default factory configuration (null configuration).
- If the service config command is present, configuration files are loaded in the following order: the network startup configuration file configured by the boot network command, the startup configuration file specified by the boot config command, /config.text, and the default factory configuration (null configuration).
- While loading configuration files in order, the system will not load another configuration files until one configuration file is loaded successfully.

Caution

The system can obtain remote files through TFTP only after you run the bootip command to configure the local IP address of the device used for initiation. Otherwise, TFTP transmission may fail during initiation. Because the system needs to use the configuration of this command in the early stage of booting, this configuration is stored in Boot ROM rather than the configuration file.

The following figure sets the boot IP address of the device and designates the network startup configuration file.

```
Ruijie(config)# boot ip 192.168.7.11
Ruijie(config)# boot network tftp://192.168.7.24/config.text
```

Configuring Preferrably Using the Network Start Configuration File

By default, the device loads the local startup configuration file specified by the **boot config** command. In some cases, if the device needs to use the network startup configuration file, run the **service config** command.

Command	Function
Ruijie(Config)# **service config**	Enables the device to preferably load the startup configuration file from the remote network server.
Ruijie(Config)#**no service config**	Disables the device to preferably load the startup configuration file from the remote network server.

This command should use in conjunction with the **boot config** and **boot network** commands.

When the device starts, the system loads the configuration file as follows;

- If the **service config** command is absent, configuration files are loaded in the following order: the startup configuration file specified by the boot config command, /config.text, the network startup configuration file configured by the boot network command, and the default factory configuration (null configuration).
- If the **service config** command is present, configuration files are loaded in the following order: the network startup configuration file configured by the boot network command, the startup configuration file specified by the boot config command, /config.text, and the default factory configuration (null configuration).
- While loading configuration files in order, the system will not load another configuration files until one configuration file is loaded successfully.

Caution

Because the system needs to use the configuration of this command in the early stage of booting, this configuration is stored in Boot ROM rather than the configuration file.

The following example loads the configuration file from a remote network server and configures the network startup configuration name.

```
Ruijie(config)# service config
Ruijie(config)# boot network tftp://192.168.7.24/config.text
```

Showing the Configuration of Start Configuration File

You can use the following command to show the configuration of a startup configuration file.

Command	Function
Ruijie# **show boot config**	Shows the configuration of a startup configuration file.
Ruijie# **show boot network**	Shows the configuration of a network startup configuration file.

The following example shows the configuration of a startup configuration file.

```
Ruijie# show boot config
Boot config file: [flash:/config_main.text]
Service config: [Disabled]
```

The following example shows the configuration of a network startup configuration file.

```
Ruijie# show boot network
Network config file: [tftp://192.168.7.24/config.text]
Service config: [Enabled]
```

Configuration Example

The following example sets up the device to first obtain the configuration file from a remote TFTP server and to use the backup configuration file in built-in flash memory when loading fails.

Step 1: Configure the device to first load the configuration file from a network server and configure the boot IP address.

```
Ruijie(config)# service config
Ruijie(config)# boot ip 192.168.7.11
```

Step 2: Configure the network startup configuration file.

```
Ruijie(config)# boot network tftp://192.168.7.24/router_1.text
```

Step 3: Configure the local startup configuration file.

```
Ruijie(config)# boot config flash:/router_1.text
```

Step 4: Show the configuration.

```
Ruijie# show boot network
Network config file: [tftp://192.168.7.24/router_1.text]
Service config: [Enabled]
Ruijie# show boot config
Boot config file: [flash:/router_1.text]
Service config: [Enabled]
```

Configuring Network Communication Detection Tools

Ping Connectivity Test

To test the connectivity of a network, many network devices support the **Echo** protocol. The protocol sends a special packet to a specified network address and waits for a response. This allows you to evaluate the connectivity, delay and reliability of a network. The ping tool provided by RGOS can effectively help users diagnose and locate the connectivity problems in a network.

The **Ping** command runs in ordinary user mode and privileged user mode. In ordinary user mode, only basic ping functions are available. However, in privileged user mode, extended ping functions are available.

Command	Function
Ruijie# **ping** [vrf *vrf-name*] [ip] [*address* [**length** *length*] [**ntimes** *times*] [data *data*][**source** *source*] [**timeout** *seconds*] [df-bit] [validate]]	Tests the network connectivity.

The basic ping function can be performed in either ordinary user mode or privileged user mode. By default, this command sends five 100-byte packets to the specified IP address. If the system receives a response within the specified time (2 seconds by default), it shows "!" . Otherwise, it shows ".". Finally, the system shows statistics. The following example shows an ordinary **ping**:

```
Ruijie# ping 192.168.5.1
Sending 5, 100-byte ICMP Echoes to 192.168.5.1, timeout is 2 seconds:
< press Ctrl+C to break >
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/10 ms
```

The extended ping function can be performed in privileged user mode only. This function allows you specify the number of packets, packet length, and timeout time. As with the basic ping function, the extended ping also shows statistics. The following example shows an extended **ping**:

```
Ruijie ping 192.168.5.197 length 1500 ntimes 100 data ffff source 192.168.4.190 timeout 3
Sending 100, 1000-byte ICMP Echoes to 192.168.5.197, timeout is 3 seconds:
  < press Ctrl+C to break >
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!
Success rate is 100 percent (100/100), round-trip min/avg/max = 2/2/3 ms
```

Ping IPv6 Connectivity Test

To test the connectivity of a network, many network devices support the **Echo** protocol. The protocol sends a special packet to a specified network address and waits for a response. This allows you to evaluate the connectivity, delay and reliability of a network. The ping tool provided by RGOS can effectively help users diagnose and locate the connectivity problems in a network.

The **Ping ipv6** command runs in ordinary user mode and privileged user mode. In ordinary user mode, only basic ping IPv6 functions are available. However, in privileged user mode, extended ping IPv6 functions are available.

Command	Function
Ruijie# **ping ipv6** [*address* **[length** *length*] [**ntimes** *times*] [**data** *data*] [**source** *source*] [**timeout** *seconds*]]	Tests the network connectivity.

The basic ping function can be performed in either ordinary user mode or privileged user mode. By default, this command sends five 100-byte packets to the specified IP address. If the system receives a response within the specified time (2 seconds by default), it shows "!" . Otherwise, it shows ".". If the response does not match the request, the system shows “C” and outputs statistics. The following example shows an ordinary **ping**:

```
Ruijie# ping ipv6 2000::1
Sending 5, 100-byte ICMP Echoes to 2000::1, timeout is 2 seconds:
< press Ctrl+C to break >
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/10 ms
```

The extended ping function can be performed in privileged user mode only. This function allows you specify the number of packets, packet length, and timeout time. As with the basic ping function, the extended ping also shows statistics. The following example shows an extended **ping**:

```
Ruijie# ping ipv6 2000::1  length 1500 ntimes 100 data ffff source 2000::2 timeout 3
Sending 100, 1000-byte ICMP Echoes to 2000::1, timeout is 3 seconds:
  < press Ctrl+C to break >
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!
Success rate is 100 percent (100/100), round-trip min/avg/max = 2/2/3 ms
```

Traceroute Connectivity Test

The **Traceroute** command is mainly used to check the network connectivity. It show all the gateways that a packet passes through from the source address to the destination address and exactly locates the fault when the network fails.

One of the network transmission rules is that the number in the TTL field in the packet will decrease by 1 every time when a packet passes through a gateway. When the number in the TTL field is 0, the gateway will discard this packet and send an address unreachable error message back to the source. According to this rule, the execution of the **traceroute** command is as follows: At first, the source sends a packet whose TTL is 1 to the destination address. The first gateway sends an ICMP error message back, indicating that this packet cannot be forwarded for TTL timeout. Then, the first gateway re-sends the packet after the TTL domain adds 1. Likewise, the second gateway returns a TTL timeout error and the process lasts until the packet reaches the destination address. By recording every address returning the ICMP TTL timeout message, you can draw the entire path passed by the IP packet from the source address to the destination address.

The **traceroute** command can run in ordinary user mode and privileged user mode. The command format is as follows:

Command	Function

Command	Function
Ruijie# **traceroute** [**vrf** *vrf-name* | **ip**] [*address* [**probe** *probe*] [**ttl** *minimum maximum*] [source *source*] [**timeout** *seconds*]]	Traces the path that a packet passes through.

The following are two examples that apply **traceroute**. In one example, network connectivity is good. In another example, some gateways in a network are not connected.

traceroute example where network connectivity is good:

```
Ruijie# traceroute 61.154.22.36
  < press Ctrl+C to break >
Tracing the route to 61.154.22.36
1      192.168.12.1      0 msec  0 msec  0 msec
2      192.168.9.2       4 msec  4 msec  4 msec
3      192.168.9.1       8 msec  8 msec  4 msec
4      192.168.0.10      4 msec  28 msec  12 msec
5      202.101.143.130   4 msec  16 msec  8 msec
6      202.101.143.154  12 msec  8 msec   24 msec
7      61.154.22.36     12 msec   8 msec   22 msec
```

As you can see, to access the host with an IP address of 61.154.22.36, the network packet passes throuth gateways 1 to 6 from the source address. Meanwhile, you can know the time that the network packet spennds to reach a gateway. This is very useful for network analysis.

a. **traceroute** example where some gateways in a network are not connected:

```
Ruijie# traceroute 202.108.37.42
< press Ctrl+C to break >
Tracing the route to 202.108.37.42
1      192.168.12.1      0 msec      0 msec      0 msec
2      192.168.9.2       0 msec      4 msec      4 msec
3      192.168.110.1     16 msec     12 msec     16 msec
4      *  *  *
5      61.154.8.129          12 msec     28 msec     12 msec
6      61.154.8.17           8 msec      12 msec     16 msec
7      61.154.8.250          12 msec     12 msec     12 msec
8      218.85.157.222        12 msec     12 msec     12 msec
9      218.85.157.130        16 msec     16 msec     16 msec
10     218.85.157.77         16 msec     48 msec     16 msec
11     202.97.40.65          76 msec     24 msec     24 msec
12     202.97.37.65          32 msec     24 msec     24 msec
13     202.97.38.162         52 msec     52 msec     224 msec
14     202.96.12.38          84 msec     52 msec     52 msec
15     202.106.192.226       88 msec     52 msec     52 msec
16     202.106.192.174       52 msec     52 msec     88 msec
17     210.74.176.158        100 msec    52 msec     84 msec
18     202.108.37.42         48 msec     48 msec     52 msec
```

As you can see, to access the host with an IP address of 202.108.37.42, the network packet passes through gateways 1 to 17 from the source address and there is failure in gateway 4.

Traceroute IPv6 Connectivity Test

The **Traceroute ipv6** command is mainly used to check the network connectivity. It shows all the gateways that a packet passes through from the source address to the destination address and exactly locates the fault when the network fails.

For network transmission rules, refer to the previous section.

The **traceroute ipv6** command can run in ordinary user mode and privileged user mode. The command format is as follows:

Command	Function
Ruijie# **traceroute ipv6** [*address* [**probe** *probe*] [**ttl** *minimum maximum*] [**source** source] [**timeout** *seconds*]]	Traces the path that a packet passes through.

The following are two examples that apply **traceroute ipv6**. In one example, network connectivity is good. In another example, some gateways in a network are not connected.

traceroute ipv6 example where network connectivity is good:

```
Ruijie# traceroute ipv6 3004::1
 < press Ctrl+C to break >
Tracing the route to 3004::1
1     3000::1       0 msec  0 msec  0 msec
2     3001::1       4 msec  4 msec 4 msec
3     3002::1       8 msec  8 msec  4 msec
4     3004::1       4 msec  28 msec  12 msec
```

As you can see, to access the host with an IP address of 3004::1, the network packet passes throuth gateways 1 to 4 from the source address. Meanwhile, you can know the time that the network packet spennds to reach a gateway. This is very useful for network analysis.

traceroute ipv6 example where some gateways in a network are not connected:

```
Ruijie# traceroute ipv6 3004::1
 < press Ctrl+C to break >
Tracing the route to 3004::1
1     3000::1       0 msec  0 msec  0 msec
2     3001::1       4 msec  4 msec 4 msec
3     3002::1       8 msec  8 msec  4 msec
4     * * *
5     3004::1       4 msec  28 msec  12 msec
```

As you can see, to access the host with an IP address of 3004::1, the network packet passes through gateways 1 to 5 from the source address and there is failure in gateway 4.Configuring File System

Configuring File System

Understanding File System

Overview

The chapter describes the file system management on RGOS. The RGOS file management offers an unified cross-platform management function, providing the unified file management interface for different kinds of devices, storages and file transmission protocols.

Locally, there are many kinds of storage medias, for instance, Universal Serial BUS (USB) and FLASH, which can be distributed on different boards like primary and secondary management boards. Users can exchange files with remote devices through xModem and TFTP protocols. These functions can be realized using the same command.

Not all types of devices and file systems support all file system commands described here, because they support different types of file operations. The Help command shows the storage media and protocols supported by the file operation commands.

Basic Features

The file system management on a RGOS device offers an unified command interface for operations of various files on the device.

Using URL to Locate a File

The file system of RGOS uses Uniform Resource Locators (URLs) to uniformly locate files and directories in storage media on the local device or remote device. For example, you can copy a file from one place to another using the **copy** *source-url destination-url* command. The destination can be on the local device or a remote server.

URL representation varies by commands. The following sections describe URL usage:

- Locate a file on the server.
- Locate a local file.
- Description of URL prefixes

Locate a file on the server

To locate a file on the server, use the following command:

- **tftp:**[[//location]/directory]/filename

location: IP address or host name. *Path (directory and file name):* position for file transmission. For instance, the file transission directory specified by the TFTP server is C:\download, the file path specified by the device is the one under C:\download. tftp://192.168.0.1/binary/rgos.bin refers to the c:\download\binary\rgos.bin file on the TFTP server with the IP address of 192.168.0.1.

⚠ Caution TFTP can only transmit files smaller than 32M. To transmit files larger than 32M, use the FTP protocol. Set a device as the FTP server to upload files to or download files from the server.

Locate a Local File

Use the [*prefix*]:[*directory/*]*filename* syntax to locate a local file on FLASH, USB and the FLASH of the management board of the device.

For example:

flash:/config.text: the configuration file on the local FLASH

usb0:/backup/rgos.bin.bak: the file on the first USB

slave:/rgos.bin: files under the root directory of the secondary management board

Note Without prefix, the syntax refers to the file system type on the current path, for instance, if the current path is under the root directory of usb0, the syntax indicates usb0 if the file system does not specify prefix.

⚠ Caution When you use a prefix to specify a local file, the path after ":" must be an absolute path.

Description of URL Prefixes

URL prefixes are used to specify file systems. Different devices and file operation commands can run different file systems. You can show the file systems supported on the device by the **show file system** command.

The following table shows the URL prefixes:

Prefix	Description
flash:	FLASH storage media, which can be used on all devices. The startup program is generally stored in the FLASH of a device when the device is delivered.
Tftp:	TFTP network server
xmodem:	Receives and sends files through xModem.
slave:	FLASH on the secondary management board of the chassis-based device
Usb0:	The first USB device
Usb1:	The second USB device
sd0:	The first SD card
sw1-m1-disk0:	Management board on the M1 slot of the chassis with switch id 1 in the VSU mode.
sw1-m2-disk0:	Management board on the M2 slot of the chassis with switch id 1 in the VSU mode.
sw2-m1-disk0:	Management board on the M1 slot of the chassis with switch id 2 in the VSU mode.
sw2-m2-disk0:	Management board on the M2 slot of the chassis with switch id 2 in the VSU mode.

Note

Different file system commands and different platforms support different types of file systems. For details, use the help information in the command line, for example:

```
Ruijier#copy ?
WORD          Copy from current file system
  flash:          Copy from flash: file system
  running-config   Copy from current system configuration
  slave:         Copy from slave: file system
  startup-config   Copy from startup configuration
  tftp:           Copy from tftp: file system
  usb0:         Copy from usb0: file system
  usb1:         Copy from usb1: file system
sd0:            Copy from sd0: file system
  xmodem:      Copy from xmodem: file system
```

Note

Given the limitation of xModem, files transmitted through xModem will be slightly larger than the real one.

Note

For chassis-based devices, the **slave:** prefix is supported and the **sw1-m1-disk0:** series prefixes are not supported in non-VSU modes, while in VSU mode, the **sw1-m1-disk0:** series prefixes are supported and the **slave:** prefix is not. In VSU mode, **copy flash:/file1 sw1-m1-disk0:/file2** and **copy sw1-m1-disk0:/file1 flash:/file2** are supported (the copy between the FLASH on the primary and secondary management boards of the VSU system), and **copy sw1-m1-disk0:/file1 sw1-m1-disk0:/file2** is also supported (the copy on the same management board: the primary or secondary management boards of the VSU system), but neither **copy sw1-m1-disk0:/file1 sw2-m2-disk0:/file2** nor **copy usb0:/file1 sw1-m1-disk0:/file2** is supported (only the combination with the **flash:** prefix is supported). In VSU mode, you can operate file systems on the primary and secondary management boards of the VSU system using commands such as **dir sw1-m1-disk0:/**. The secondary management board on VSU master and slave chassis can only be used to increase bandwidth. Currently, you cannot operate file systems on the secondary management boards of VSU master and slave chassis on the primary management board of the VSU system.

Showing the File System Information

This command shows all the file systems supported on the device and their available spaces.

In privileged EXEC mode, use the following command:

```
Ruijie#show file systems


File Systems:
       Size(b)       Free(b)       Type     Flags     Prefixes
   ------------   ------------   ---------   -------   ----------
```

```
*      33488896      16191488      flash        rw       flash:
             -             -       flash       rw        usb0:
             -             -       flash       rw        usb1:
             -             -       flash       rw         sd0:
             -             -       flash       rw       slave:
             -             -     network        rw       tftp:
             -             -     network        rw      xmodem:
--------------------------------------------------------------------
```

In this informatin, “*” means the active file system, **Size** means the space of the file system and **Free** means the available space.

Caution

Free means the idle status of the file system, not the size of files that can be stored. Since the file system has its own management overhead, the size of files that the system finally can store is slightly smaller than the free space.

Managing Local Files

Local files refer to ones stored in various storage media on the device, for instance, FLASH, and USB. System files such as main program, configuration files, logs and web files are stored generallly in FLASH. Some devices come with USB interfaces. Management of files on the flash disk is also local file management. For a chassis-based device with two management boards, you can manage files in the FLASH of the secondary management board with the **slave** prefix of URL.

For local files, you can:

- Copy files
- Move files
- Delete files
- Create directories
- Delete directories
- Show directories
- Show the current working path
- Modify the working path

These operations apply to slave-, USB-, or FLASH-type file systems and can copy files between these file systems.

Caution

File name is case sensitive on the FLASH- and slave- file systems. For example, abc.txt and Abc.txt are different. The file name must be entered correctly to locate the corresponding file. On USB-type file system, however, file name is not case sensitive, namely abc.txt and Abc.txt are considered the same.

Caution

Number and size of files will affect the startup and operation speed of files considerably. Storing too many or large files in FLASH will slow down startup of the device and update of the system. When the device starts for the first time, using the **dir** command will result in longer waiting time. Generally, we recommend a file system space of less than 128M. When necessary, we recommend storing a large number of files on a flash disk. We recommend clearing some old and useless files manually on a regular basis.

For chassis-based devices, a timeout failure may occur when the file system on the secondary management board (or the secondary management board of the VSU system in the VSU mode) is operated if too many files are stored on the secondary management board, the device is just started or for the first time. In such case, wait for a while according to the prompt and try again later.

Caution

Some files are vital for the system to work properly. Deleting these files will cause malfunction. These system files include:

- RCMS configuration file (/rcms_config.ini)
- Web management package (/web_management_pack.upd)
- Main program files (for multi-boot-supported devices, the main program includes all the files in the boot system configuration)

The system will automatically recognize these files and alarm you before you delete them. If you need to delete system files, the system will print WARN-level logs as below:

Ruijie# **delete** *rgos.bin*
File [rgos.bin] is a system file. System may not work properly without it.
Are you sure you want to delete it? [no] yes
0:1:1:38 Ruijie: FS-4-SYSTEM_FILE_DELETED: System file [rgos.bin] deleted!

Note

The file name with a path should be no more than 4096 bytes. Wildcard is not supported for file name and path.

Transmitting Files through Communication Protocols

- Transmitting files through TFTP:

You can upload and download files to and from the TFTP server.

In CLI privileged EXEC mode, use the following command to download files:

```
Ruijie# copy tftp:[[//location]/directory]/filename destnation-url
```

In CLI privileged EXEC mode, use the following command to upload files:

```
Ruijie# copy source-url  tftp:[[//location]/directory]/filename
```

- Transmitting files through xModem:

In CLI privileged EXEC mode, use the following command to download files:

```
Ruijie# copy xmodem:  destnation-url
```

In CLI privileged EXEC mode, use the following command to upload files:

```
Ruijie# copy source-url xmodem:
```

Working Principles

None

Protocols Specifications

None

Default Configuration

None

Typical Configuration Example

Downloading Files from TFTP Server

The following example shows how to download a.dat from directory c:\download\ of the TFTP server to the local device:

1) Run the TFTP Server on the host and select C:\download where the file to be downloaded is located.

b. Use the **ping** command to test the connection between the device and the TFTP server.

c. Log on the device, enter privileged EXEC mode and run the following command:

```
Ruijie#copy tftp://192.168.201.54/a.dat flash:
Destination filename [a.dat]?
Accessing tftp://192.168.201.54/a.dat
!!!!!!!!!!!!!!!!!!!!!!
Transmission finished, file length 343040
```

d. Run the **dir** command to show the files on the device.

```
Ruijie#dir
Directory of flash:/
   Mode Link     Size             MTime Name
-------- ---- --------- ------------------- ------------------
          1    343040 2009-01-01 02:02:59 a.dat
          1  10838016 2009-01-01 00:08:38 rgos.bin
          1       399 2009-01-01 00:01:37 config.text
--------------------------------------------------------------
3 Files (Total size 11181455 Bytes), 9 Directories.
Total 33030144 bytes (31MB) in this device, 20492288 bytes (19MB) available.
```

Uploading Files to TFTP Server

The following example shows how to upload a.dat to the c:\download\ of TFTP server:

1) Run the TFTP Server on the host and select directory C:\download where the file to be uploaded is located.

e. Use the **ping** command to test the connection between the device and the TFTP server.

f. Log on the device, enter privileged EXEC mode and run the following command:

```
Ruijie#copy flash:/a.dat tftp://192.168.201.54/a.dat
Accessing flash:a.dat...
!!!!!!!!!!!!!!!!!!!!!!!
 Transmission finished, file length 343040
```

g. Check whether a.dat exists under C:\download on TFTP server.

Downloading Files through xModem

The following example shows how to download config.txt from PC to the local device through xModem:

1) Use a serial cable to connect the seiral interface of PC to the serial interface of the device.

h. Run hyperterminal of Windows to connect to the console of the device.

i. In privileged EXEC mode, use the following command to download the file:

```
Ruijie# copy xmodem: flash:/config.text
```

j. In the Windows hyperterminal of local deivce, select Transmit files of Transmit menu:

k. In the pop-up dialog box, select the file to download and xModem and click Transmit. The Windows hyperterminal shows the transmission progress and packets.

l. Run the **dir** command to show the files on the device.

```
Directory of flash:/
   Mode Link      Size             MTime Name
-------- ---- --------- ------------------- -------------------
          1    343040 2009-01-01 02:02:59 a.dat
          1  10838016 2009-01-01 00:08:38 rgos.bin
          1       399 2009-01-01 00:01:37 config.text
---------------------------------------------------------------
3 Files (Total size 11181455 Bytes), 0 Directories.
Total 33030144 bytes (31MB) in this device, 20492288 bytes (19MB) available.
```

Uploading Files through xModem

The following example shows how to upload config.txt from the local device to C:\Documents and Settings\ju of PC through xModem:

1) Use a serial cable to connect the serial interface of PC and the serial interface of the device.

m. Run the hyperterminal of Windows to connect to the console of the device.

n. In privileged EXEC mode, use the following command to upload file:

```
Ruijie# copy flash:/config.text xmodem
```

o. In the Windows hyperterminal of local deivce, select Receive files of Transmit menu:

p. In the pop-up dialog box, select the directory to save the uploaded file and xModem. Click Receive. The Windows hyperterminal prompts to set the name used to store the file. Click OK.

q. Check whether config.text exists under C:\Documents and Settings\ju on PC.

Moving Files from FLASH to USB Device

The following example shows how to move config.txt from FLASH to flash disk on USB0 and save it in the **backup** directory of flash disk:

```
Directory of flash:/
   Mode Link     Size            MTime     Name
-------- ---- --------- ------------------- ------------------
          1    343040 2009-01-01 02:02:59   a.dat
          1  10838016 2009-01-01 00:08:38   rgos.bin
          1      399  2009-01-01 00:01:37    config.text
--------------------------------------------------------------
3 Files (Total size 11181455 Bytes), 0 Directories.
Total 33030144 bytes (31MB) in this device, 20492288 bytes (19MB) available.
```

Enter the root directory of flash disk:

```
Ruijie#cd usb0:/
```

Confirm the current path:

```
Ruijie#pwd usb0:/
```

Create **backup** directory on flash disk:

```
Ruijie#mkdir backup
```

Copy the file to the flash disk:

```
Ruijie#copy flash:/config.text config.text
```

Check the result.

```
Ruijie#dir backup
Directory of usb0:/backup
   Mode Link     Size            MTime     Name
-------- ---- --------- ------------------- ------------------
          1      399   2009-01-01 00:01:37  config.text
--------------------------------------------------------------
Total 33030144 bytes (31MB) in this device, 20488192 bytes (19MB) available.
```

Moving Files from FLASH to SD Card

The following example shows how to move config.txt from FLASH to SD card and save it in the **backup** directory of SD card:

```
Directory of flash:/
   Mode Link     Size            MTime     Name
-------- ---- --------- ------------------- ------------------
          1    343040 2009-01-01 02:02:59   a.dat
          1  10838016 2009-01-01 00:08:38   rgos.bin
          1      399  2009-01-01 00:01:37    config.text
--------------------------------------------------------------
3 Files (Total size 11181455 Bytes), 0 Directories.
Total 33030144 bytes (31MB) in this device, 20492288 bytes (19MB) available.
```

Enter the root directory of SD card:

```
Ruijie#cd sd0:/
```

Confirm the current path:

```
Ruijie#pwd sd0:/
```

Create **backup** directory on the SD card:

```
Ruijie#mkdir backup
```

Make sure that the directory is created:

```
Ruijie#dir
Directory of sd0:/
   Mode Link     Size             MTime     Name
-------- ---- --------- ------------------- ------------------
   <DIR>   1    343040 2009-01-01 02:02:59    backup
--------------------------------------------------------------
3 Files (Total size 11181455 Bytes), 0 Directories.
Total 33030144 bytes (31MB) in this device, 20492288 bytes (19MB) available.
```

Copy the file to the SD card:

```
Ruijie# copy flash:/config.text backup/config.text
```

Check the result:

```
Ruijie#dir backup
Directory of sd0:/backup
   Mode Link     Size             MTime     Name
-------- ---- --------- ------------------- ------------------
          1      399   2009-01-01 00:01:37  config.text
--------------------------------------------------------------
Total 33030144 bytes (31MB) in this device, 20488192 bytes (19MB) available.
```

Copying Files between USB and SD Card

The following example shows how to copy rgos_10_4.bin from flash disk to SD card:

Check the available space on the SD card:

```
Ruijie#dir sd0:/
Directory of sd0:/
   Mode Link     Size             MTime Name
```

```
-------- ---- --------- ------------------- ------------------
   <DIR>    2         0 2035-02-11 23:24:34 backup/
            1   7650112 2035-02-11 23:42:25 rgos.bin
--------------------------------------------------------------
1 Files (Total size 7650112 Bytes), 1 Directories.
Total 528482304 bytes (504MB) in this device, 475058176 bytes (453MB) available.
```

Copy the file from flash disk to SD card:

```
Ruijie#copy usb0:/rgos_10_4.bin sd0:/rgos_10_4.bin
[OK 7,650,112 bytes]
```

Check the result:

```
Ruijie#dir sd0:/
Directory of sd0:/
    Mode Link      Size               MTime Name
-------- ---- --------- ------------------- ------------------
   <DIR>    2         0 2035-02-11 23:24:34 backup/
            1   7650112 2035-02-11 23:42:25 rgos.bin
            1   7650112 2035-02-11 23:47:36 rgos_10_4.bin
--------------------------------------------------------------
2 Files (Total size 15300224 Bytes), 1 Directories.
Total 528482304 bytes (504MB) in this device, 459571200 bytes (438MB) available.
```

Copy the file from SD card to flash disk:

```
Ruijie#copy sd0:/rgos_10_4.bin usb0:/new_rgos.bin
[OK 7,650,112 bytes]
```

Check the result:

```
Ruijie#dir usb0:/
Directory of usb0:/
    Mode Link      Size               MTime Name
-------- ---- --------- ------------------- ------------------
            1   7650112 2035-02-11 23:49:21 new_rgos.bin
            1   7650112 2035-02-11 23:45:42 rgos_10_4.bin
--------------------------------------------------------------
2 Files (Total size 15300224 Bytes), 0 Directories.
Total 528482304 bytes (504MB) in this device, 451784704 bytes (430MB) available.
```

Copying Files from Primary Management Board to Secondary Management Board

The following example shows how to copy rgos_10_4.bin from primary management board to secondary management board:

Check the FLASH space on secondary management board:

```
Ruijie#dir slave:/
```

```
Directory of slave:/
   Mode Link     Size             MTime     Name
-------- ---- --------- ------------------- ------------------
           1   11014633  2006-01-01 08:00:46  rgos.bin
           1    399      2006-01-01 08:01:37  config.text
--------------------------------------------------------------
2 Files (Total size 11181455 Bytes), 0 Directories.
Total 33030144 bytes (31MB) in this device, 20488192 bytes (19MB) available.
```

Copy rgos_10_4.bin from primary management board to the secondary one:

```
Ruijie#copy rgos_10.4.bin slave:/
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! [ok 10,234,345 bytes]
```

Check the result:

```
Ruijie#dir slave:/
Directory of slave:/
   Mode Link     Size                MTime     Name
-------- ---- --------- ------------------- ------------------
          1     11014633   2006-01-01 08:00:46  rgos.bin
          1     11025788   2008-01-01 08:00:46  rgos_10.4.bin
          1     399        2006-01-01 08:01:37  config.text
--------------------------------------------------------------
3 Files (Total size 11181455 Bytes), 0 Directories.
Total 33030144 bytes (31MB) in this device, 9563693 bytes (9MB) available
```

Deleting Directories

The following example shows how to delete a non-empty aaa directory:

Show the current directory status:

```
Ruijie#dir
Directory of flash:/
   Mode Link     Size                MTime     Name
-------- ---- --------- ------------------- ------------------
          1      11014633 2006-01-01 08:00:46  rgos.bin
 <dir>    1      0        2006-01-01 08:00:00 aaa/
          1      399      2006-01-01 08:01:37 config.text
--------------------------------------------------------------
2Files (Total size 11015032 Bytes), 1 Directories
Total 33030144 bytes (31MB) in this device, 9563693 bytes (9MB) available
```

Check whether there is any file under the directory:

```
Ruijie#dir aaa
```

```
Directory of flash:/aaa
   Mode Link     Size                MTime     Name
-------- ---- --------- ------------------- -------------------
         1         149  2006-01-01 08:01:37 backup.txt
--------------------------------------------------------------
1Files (Total size 149 Bytes), 0 Directories
Total 33030144 bytes (31MB) in this device, 9563693 bytes (9MB) available
```

Delete a non-empty directory:

```
Ruijie# delete recursive aaa
```

Delete an empty directory:

```
Ruijie# rmdir aaa
```

Note

The devices locating files through URL such as S86 and S12000 distributed devices support URL parameters (to locate files) and does not support deleting the non-null directory recursively (recursive parameters are not supported)

Check the result:

```
Ruijie#dir
Directory of flash:/
   Mode Link     Size                MTime     Name
-------- ---- --------- ------------------- -------------------
         1     11014633 2006-01-01 08:00:46  rgos.bin
         1     399     2006-01-01 08:01:37 config.text
--------------------------------------------------------------
2Files (Total size 11015032 Bytes), 0 Directories
Total 33030144 bytes (31MB) in this device, 9563693 bytes (9MB) available
```

Configuring Syslog

Overview

During the operation of a device, there are various state changes, such as the link status up/down, and various events occurring, such as receiving abnormal messages and handling exceptions. Our product provides a mechanism to generate messages of fixed format (log message) in case of status change or event occurring. These messages can be displayed in related windows (console, VTY, etc.) or recorded in related media (memory buffer, FLASH), or sent to a group of log servers in the network for the administrators to analyze and locate problems. Meanwhile, in order to make it easy for administrators to read and manage log messages, these log messages can be labeled time stamps and serial numbers, and is graded according to the priority of log information.

Log Message Format

The format of the our log message is as follows:

```
<priority> seq no: timestamp sysname: %severity
%ModuleName-severity-MNEMONIC: description
```

Their meanings are as follows:

Command	Meaning
<priority>	Priority, priority value = Device value x 8 + Severity
seq no	System serial number, a six-digit integer. You can disable this information output by commands.
timestamp	Timestamp, local time by default. In the format of Mmm dd hh:mm:ss, Mmm indicates the English abbreviations of the 12 months. From January to December, they are abbreviated as: Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, and Dec.
sysname	System name, you can disable this information output by commands.
ModuleName	Abbreviation of functional module name.
severity	Log severity level.
MNEMONIC	Information abbreviation
description	Information description

For example:

```
<189> 226:Mar  5 02:09:10 Ruijie %SYS-5-CONFIG_I: Configured from console by console
```

Caution

The priority field is not attached to the log messages that are printed in the user window. It only appears in the log messages that are sent to the syslog server.

Log Configuration

Log Switch

The log switch is turned on by default. If it is turned off, the device will not print log information in the user window, or send log information to the syslog server, or record the log information in the related media (memory buffer, flash).

To turn on or off the log switch, run the following command in global configuration mode:

Command	Function
Ruijie(config)# **logging on**	Turns on the log switch
Ruijie(config)# **no on**	Turns off the log switch

Caution Do not turn off the log switch in general case. If it prints too much information, you can reduce it by setting different displaying levels for device log information.

Configuring the Device for Displaying the Log Information

When the log switch is turned on, the log information will be displayed on the console and also sent to different displaying devices. To configure different displaying devices to receive logs, run the following commands in global configuration mode or privileged user level:

Command	Function
Ruijie(config)# **buffered** [*buffer-size* | *level*]	Records log in memory buffer
Ruijie# **termninal monitor**	Allows log to be displayed on VTY window
Ruijie(config)# **logging server** *host*	Sends log information to the syslog sever on the network
Ruijie(config)# **logging file flash:***filename* [*max-file-size*] [*level*]	Records log on extended FLASH. This command creates a file based on the specified file name on the FLASH to store logs. The file size increases with the log size, but its upper limit is subject to the configured max-file-size.

Buffered will record log information in the memory buffer. The memory buffer for log is used in recycled manner. That is, when it is full, the oldest information will be overwritten. To show the log information in the memory buffer, run the **show logging** command at privileged user level. To clear the log information in the memory buffer, run the **clear logging** command at privileged user level.

Terminal monitor allows log information to be displayed on the current VTY (such as the telnet window).

Logging server host specifies the address of the syslog server that will receive the log information. Our product allows the configuration of at most 5 syslog servers. The log information will be sent to all the syslog servers at the same time. The configuration of the **logging host** command has the same results.

Caution To send the log information to the syslog server, it is required to turn on the timestamp switch or serial number switch of the log information. Otherwise, log information will not be sent to the syslog server.

Logging file flash: Records log information in FLASH. The filename for log shall not have any extension to indicate the file type. The extension of the log file is fixed as txt. Any configuration of extension for the filename will be refused.

More flash: The **filename** command shows the contents of the log file in the flash.

Caution

Some devices support extended FLASH. If the device has extended FLASH, the log information will be recorded there. If the device has no extended FLASH, the log information will be recorded in the serial FLASH.

Enabling the Log Timestamp Switch of Log Information

To add or delete timestamp in log information, run the following command in global configuration mode:

Command	**Function**
Ruijie(config)# **service timestamps** [*message-type* [**uptime** | **datetime** [**msec**] [**year**]]]	Enables the timestamp in the log information
Ruijie(config)# **no service timestamps** [*message-type*]	Disables the timestamp in the log information

The timestamp are available in two formats: device uptime and device datetime. Select the type of timestamp as required.

Message type: log or debug. The "log" type means the log information with severity levels 0-6. The "debug" type means that with severity level 7.

Caution

If the current device has no RTC, the configured time is invalid, and the device automatically uses the startup time as the timestamp for the log information. If the device has an RTC, the device time is used as the timestamp for the log information by default.

Enabling System Name Switches in Log System

By default, the system name is not included in the log information. To add or remove the system name in the log information, run the following commands in global configuration mode.

Command	**Function**
Ruijie(config)# **no service sysname**	Removes the system name in the log message.
Ruijie(config)# **service sysname**	Adds a system name to the log message.

Enabling Log Statistics

By default, the log statistics function is disabled. To enable or disable the log statistics function, run the following commands in global configuration mode.

Command	**Function**

Ruijie(config)# **no logging count**	Disables the log statistics function and delete the statistics about the log information.
Ruijie(config)# **logging count**	Enables the log statistics function.

Ruijie# show logging count

```
Module Name   Message Name     Sev Occur   Last Time
================================================================
LINEPROTO     UPDOWN            5   2      Aug 20 01:41:19
-------------------------------------------------------------
LINEPROTO TOTAL                        2
LINK         CHANGED            5   1      Aug 20 01:41:19
-------------------------------------------------------------
LINK TOTAL                             1
SYS          CONFIG_I           5   1      Aug 20 01:40:55
-------------------------------------------------------------
SYS TOTAL                              1
```

Ruijie #**config**

```
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie (config)#no logging count
Ruijie (config)#end
Ruijie #show logging count
Module Name   Message Name          Sev Occur   Last Time
===========================================================
```

Enabling the Serial Number Switch of Log Information

By default, the log information has no serial number. To add or delete the serial number in log information, run the following command in global configuration mode:

Command	Function
Ruijie(config)# **no service sequence-numbers**	Deletes the serial number in the log messages.
Ruijie(config)# **service sequence-numbers**	Adds a serial number to the log messages.

Note

The log serial number is a long integer, which increases in ascending order when a log is added. However, since only five digits of the serial number is displayed, when it increases from 1 to 100000 or reaches 2^32, a turnover occurs, that is the serial number is displayed from 00000 again.

Configuring Synchronization Between User Input and Log Output

By default, user input is asynchronous with log output. User input is interrupted if the log is output when the user is keying in characters. As following shows, the status of FastEthernet 0/12 changes and a log is printed after the user entered **vlan**, so that the user forgot which character he was entering previously, affecting the coherence of command entering.

```
Ruijie(config)#vlan Aug 20 16:46:49 %LINK-5-CHANGED: Interface FastEthernet 0/12, changed
state to down
Aug 20 16:46:49 %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet 0/12, changed
state to DOWN
% Incomplete command.
```

While after the synchronization function is configured, the contents that the user entered previously will be displayed even though a log is printed when the user is entering a command, ensuring integrity and coherence. As following shows, the status of FastEthernet 0/1 changes and a log is printed after the user entered **vlan**, but the log module automatically prints **vlan** after the log is printed for the user to continue.

```
Ruijie(config)#vlan
*Aug 20 10:05:19: %LINK-5-CHANGED: Interface GigabitEthernet 0/1, changed state to up
*Aug 20 10:05:19: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet 0/1,
changed state to up
Ruijie(config)#vlan
To configure synchronization between user input and log output, run the following commands
in line configuration mode:
```

Command	Function
Ruijie(config-line)# **logging synchronous**	Sets synchronization between user input and log output.
Ruijie(config)# **no logging synchronous**	Removes synchronization between user input and log output.

Configuring Log Rate Limit

By default, log rate is not limited. However, when there are massive logs, no log rate limit will cause burden on the system. To configure log rate limit, run the following commands in global configuration mode:

Command	Function
Ruijie(config)# **logging rate-limit** *number*	Sets log rate limit on the log information.
Ruijie(config)# **no logging rate-limit**	Removes the setting of log rate limit.

Configuring the Log Information Displaying Level

Users can set the severity level of log information that is allowed to be displayed to view the log information of a specific severity level.

To configure the log information displaying level, run the following commands in global configuration mode:

Command	Function
Ruijie(config)# **logging console** [*level*]	Sets the level of log information that is allowed to be displayed on the console.
Ruijie(config)# **logging monitor** [*level*]	Sets the level of log information that is allowed to be displayed on the VTY window (such as telnet window).
Ruijie(config)# **logging buffered** [*buffer-size*] [*level*]	Sets the level of log information that is allowed to be recorded in memory buffer.

Ruijie(config)# **logging file flash:***filename* [*max-file-size*] [*level*]	Sets the level of log information that is allowed to be recorded in extended flash.
Ruijie(config)# **logging trap** [*level*]	Sets the level of log information that is allowed to be sent to syslog server.

The log information of Ruijie Networks products is classified into the following 8 levels:

Level Keyword	Level	Description
Emergencies	0	Emergency case, system cannot run normally
Alerts	1	Problems that need immediate remedy
Critical	2	Critical conditions
Errors	3	Error message
Warnings	4	Alarm information
Notifications	5	Information that is normal but needs attention
Informational	6	Descriptive information
Debugging	7	Debugging messages

Lower value indicates higher level. That is, level 0 indicates the information of the highest level.

When the level of log information that can be displayed is set for the specified device, the log information that is at or below the set level will be displayed. For example, after the **logging console *6*** command is executed, all log information at or below level 6 will be displayed on the console.

The log information that is allowed to be displayed on the console is at level 7 by default.

The log information that is allowed to be displayed on the VTY window is at level 7 by default.

The log information that is allowed to be sent to the syslog server is at level 6 by default.

The log information that is allowed to be recorded in the memory buffer is at level 7 by default.

The log information that is allowed to be recorded in the extended flash is at level 6 by default.

You can use the **show logging** command in privileged mode to show the level of log information allowed to be displayed on different devices.

Configuring the Device Value of the Log Information

The device value is one of the parts that form the priority field in the messages sent to the syslog server, indicating the type of device that generates the information.

To configure the log information device value, run the following command in global configuration mode:

Command	Function
Ruijie(config)# **logging facility** *facility-type*	Configures the device value of the log information.
Ruijie(config)# **no logging facility** *facility-type*	Restores the device value of the log information to the default value.

The meanings of various device values are described as below:

```
Numerical Code       Facility
         0            kernel messages
         1          user-level messages
         2          mail system
         3          system daemons
         4          security/authorization messages
         5          messages generated internally by syslogd
         6          line printer subsystem
         7          network news subsystem
         8          UUCP subsystem
         9          clock daemon
        10          security/authorization messages
        11          FTP daemon
        12          NTP subsystem
        13          log audit
        14          log alert
        15          clock daemon
        16          local use 0  (local0)
        17          local use 1  (local1)
        18          local use 2  (local2)
        19          local use 3  (local3)
        20          local use 4  (local4)
        21          local use 5  (local5)
        22          local use 6  (local6)
        23          local use 7  (local7)
```

The default device value of Ruijie products is 23.

Configuring the Source Address of Log Messages

By default, the source address of the log messages sent to the syslog server is the address of the port that sends the messages. You can fix the source address for all log messages through commands.

It is possible to directly set the source IP address of the log messages or the source port of the log messages.

To configure the source address of the log messages, run the following command in global configuration mode:

Command	Function
Ruijie(config)# **logging source interface** *interface-type interface-number*	Configures the source port of log information.
Ruijie(config)# **logging source** {**ip** *ip-address* | **ipv6** *ipv6-address*}	Configures the source IP address of log messages.

Note

If the configured source IP address of the log message is not configured on any interface of the device, the source IP address of the log message is this inexistent address. However, it is not recommended to perform such configuration in actual use.

Setting and Sending User Log

Command	Function
Ruijie(config)# **logging userinfo**	Sets user login/logoff log.
Ruijie(config)# **logging userinfo command-log**	Send a log when a configuration command is executed.

Log Monitoring

To monitor log information, run the following commands in privileged user mode:

Command	Function
Ruijie# **show logging**	Views the log messages in memory buffer as well as the log statistics.
Ruijie# **show logging count**	Views the statistics of logs in every module.
Ruijie# **clear logging**	Clears the log messages in the memory buffer.
Ruijie# **more flash:***filename*	Views the log files in the extended flash.

⚠ Caution

The format of the timestamp in the output result of the **show logging count** command is the format in the latest log output.

Examples of Log Configurations

Here is a typical example to enable the logging function. Connect the device to the log server, whose IP address is 192.168.200.2. Perform the following configuration to make all logs carry timestamps and allow logs of all levels to be sent to the log server:

```
Ruijie(config)# service timestamps debug datetime   //Enable debug information timestamp, in date format
Ruijie(config)# service timestamps log datetime    //Enable log information timestamp, in date format
Ruijie(config)# logging 192.168.200.2      //Specify the syslog server address
Ruijie(config)# logging trap debugging     //The log information of all levels will be sent to the syslog server
Ruijie(config)# end
```

Configuring Device Fault Management

Overview of Device Fault Management Module

Purpose

The device fault management module manages device faults, which generates alarms to alert router faults, protects devices against exceptions and adds some methods of preventing faults, such as displaying working status of various basic hardware devices, to enhance safe router running level.

Requirements and Notes

Hardware Requirements

Because fault alarms are closely related to hardware, many functions of the device fault management module are directly related to hardware. For example, hardware must support the detection of power voltage.

Caution

The device fault management module version 1.0 (DFM1.0) cannot display power voltage, fan performance and operating inlet temperature, and fails to allow MIB to search for this information.

System Support

Not all devices support alarm generation by interruption in the case of a device fault. As a result, scheduled detection is required. Here, the detection results obtained last time are used as displayed information and detection interval is set to five seconds that cannot be modified by users.

Running Mode

At present, all configuration of the device fault management module is carried out in privilege mode. As a result, to run the commands mentioned in this document, enter privilege mode first.

Checking Status Information

The command tree of status displaying of the total fault management:

Command	Function
show environment [alarms | all | fans | hardware | powers]	Displays environment of the managed faulty device.

Displaying Exception Alarm

Command	Function
show environment alarms	Displays information about alarm processing. For example, fans should be checked in the case of excessively high temperature

The following information is displayed for this command:

```
Ruijie# show environment alarms
Warning!!!
Power supplies have been changed since the router start, please check them
Warning!!!
Fans have been changed since the router start, please check them.
Warning!!!
Temperature is high, please check powers and fans.
Ruijie#
```

Displaying Operating Temperature

Command	Function
show temperature	Displays temperature of the current operating environment, namely temperature inside the chassis. Currently, inlet temperature cannot be detected.

The following information is displayed for this command:

```
Ruijie#show temperature
Device    Temperature(C)
------    -----------
 CM          43
```

Displaying Fan Information

Command	Function
show environment fans	Displays running situations and status information on one or multiple fans, including whether fans run normally and the number of fans. Currently, performance detection is not supported.

The following information is displayed for this command:

```
Ruijie# show environment fans
Environmental status update at 11:31:37 Jan 9, 1944.
Data is 13 second old, refresh in 20 second(s).
Fans working status:
Fan 0 is on.
Fan 1 is on.
Fan 2 is on.
Fan 3 is on.
Fan 4 is on.
Fan 5 is on.
Fan 6 is on.
Fan 7 is on.
```

Displaying Power Supply Information

Command	Function
show environment powers	Displays the current status information on the power supply, including rated operating voltage, the number of power supplies, and whether each power supply works normally. At present, the current operating voltage and threshold cannot be detected.

The following information is displayed for this command:

```
Ruijie# show environment powers
Environmental status update at 11:28:50 Jan 9, 1944.
Data is 10 second old, refresh in 20 second(s).
Power Supplies:
Power supply 1 is present. Unit is on.
Power supply 2 is present. Unit is on.
Power supply 3 is present. Unit is on.
```

Displaying Information Related to Hardware

Command	Function
show environment hardware	Displays the current status information on the hardware, including CPU name and speed.

The following information is displayed for this command:

```
Ruijie#show environment hardware
  Environmental status update at 16:25:26 2011-01-20.
  Data is 13 second old, refresh in 20 second(s).
  Hardware:
        CPU name: BCM1250.
        CPU Speed : 800M
```

Displaying All Information on Fault Management

Command	Function
show environment all	Displays device status information in the current device fault management.

The following information is displayed for this command:

```
Ruijie#show environment all
  Environmental status update at 16:26:46 2011-01-20.
  Data is 18 second old, refresh in 20 second(s).
  Power Supplies:
        Power supply 1 is not present. Unit is off.
        Power supply 2 is present. Unit is on.
        Power supply 3 is not present. Unit is off.
  Fans working status:
        Fan 1 is on.
```

```
      Fan 2 is on.
      Fan 3 is on.
      Fan 4 is on.
      Fan 5 is on.
      Fan 6 is on.
 Hardware:
      CPU name: BCM1250.
      CPU Speed : 800M
```

Configuring SNMP

SNMP Overview

Introduction

As the abbreviation of Simple Network Management Protocol, SNMP has been a network management standard (RFC1157) since the August, 1988. So far, the SNMP becomes the actual network management standard for the support from many manufacturers. It is applicable to the situation of interconnecting multiple systems from different manufacturers. Administrators can use the SNMP protocol to query information, configure network, locate failure and plan capacity for the nodes on the network. Network supervision and administration are the basic function of the SNMP protocol.

As a protocol in the application layer, the SNMP protocol works in the client/server mode, including three parts as follows:

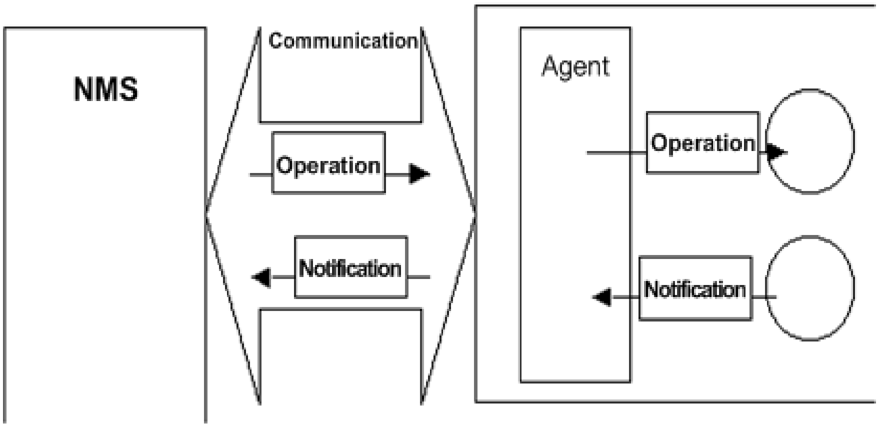
- SNMP network manager
- SNMP agent
- MIB (management information base)

The SNMP network manager, also referred to as NMS (Network Management System), is a system to control and monitor the network using the SNMP protocol.

The SNMP Agent is the software running on the managed devices. It receives, processes and responds the monitoring and controlling messages from the NMS, and also sends some messages to the NMS.

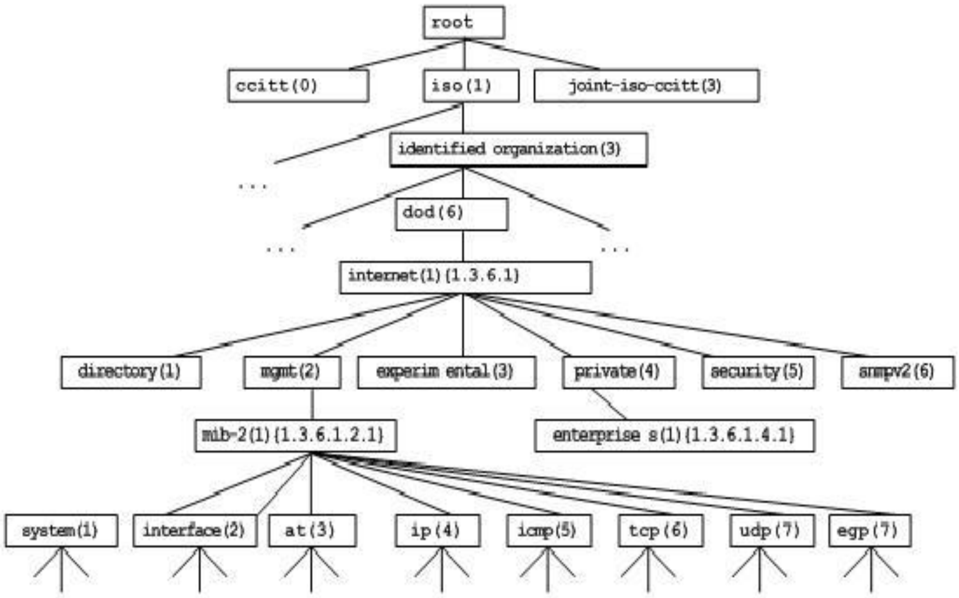
The relationship between the NMS and the SNMP Agent can be indicated as follows:

Figure 1 Relationship between the NMS and the SNMP Agent



The MIB (Management Information Base) is a virtual information base for network management. There are large volumes of information for the managed network equipment. In order to uniquely identify a specific management unit in the SNMP message, the tree-type hierarchy is used to by the MIB to describe the management units in the network management equipment. The node in the tree indicates a specific management unit. Take the following figure of MIB as an example to name the objectives in the tree. To identify a specific management unit **system** in the network equipment uniquely, a series of numbers can be used. For example, the number string {1.3.6.1.2.1.1} is the object identifier of a management unit, so the MIB is the set of object identifiers in the network equipment.

Figure 2 Tree-type MIB hierarchy



SNMP Versions

This software supports these SNMP versions:

- SNMPv1: The first formal version of the Simple Network Management Protocol, which is defined in RFC1157.
- SNMPv2C: Community-based Administrative Framework for SNMPv2, an experimental Internet protocol defined in RFC1901.
- SNMPv3: Offers the following security features by authenticating and encrypting packets:

r. Ensure that the data are not tampered during transmission.

s. Ensure that the data come from a valid data source.

t. Encrypt packets to ensure the data confidentiality.

Both the SNMPv1 and SNMPv2C use a community-based security framework. They restrict administrator’s operations on the MIB by defining the host IP addresses and community string.

With the GetBulk retrieval mechanism, SNMPv2C sends more detailed error information type to the management station. GetBulk allows you to obtain all the information or a great volume of data from the table at a time, and thus reducing the times of request and response. Moverover, SNMPv2C improves the capability of handing errors, including expanding error codes to distinguish different kinds of errors, which are represented by one error code in SNMPv1. Now, error types can be distinguished by error codes. Since there may be the management workstations supporting SNMPv1 and SNMPv2C in a network, the SNMP agent must be able to recognize both SNMPv1 and SNMPv2C messages, and return the corresponding version of messages.

SNMP Management Operations

For the information exchange between the NMS and the SNMP Agent, six types of operations are defined:

u. Get-request: The NMS gets one or more parameter values from the SNMP Agent.

v. Get-next-request: The NMS gets the next parameter value of one or more parameters from the SNMP Agent.

w. Get-bulk: The NMS gets a bulk of parameter values from the SNMP Agent.

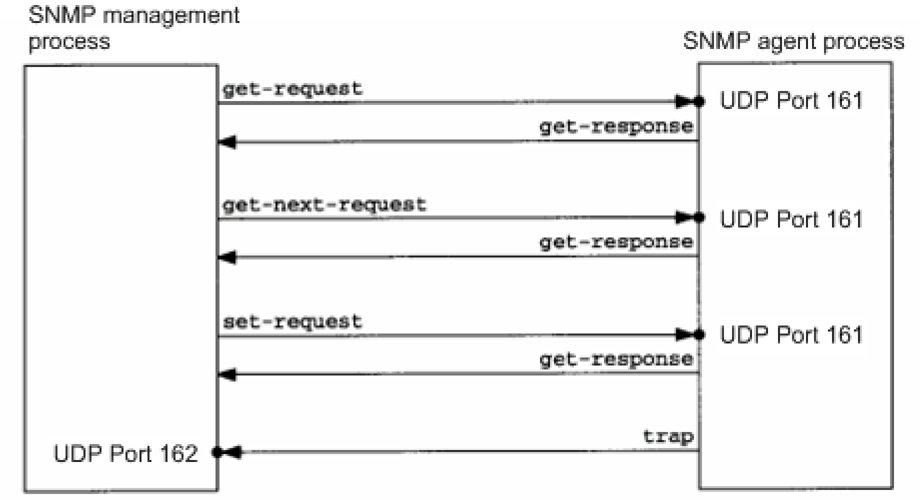
x. Set-request: The NMS sets one or more parameter values for the SNMP Agent.

y. Get-response: The SNMP Agent returns one or more parameter values, the response of the SNMP Agent to any of the above 3 operations of the NMS.

z. Trap: The SNMP Agent proactively sends messages to notify the NMS that some event will occur.

The first four messages are sent from the NMS to the SNMP Agent, and the last two messages are sent from the SNMP Agent to the NMS (Note: SNMPv1 does not support the Get-bulk operation). These operations are described in the following figure:

Figure 3 Message types in SNMP



NMS sends messages to the SNMP Agent in the first three operations and the SNMP Agent responds a message through the UDP port 161. However, the SNMP Agent sends a message in the Trap operation through the UDP port 162.

Caution

When managing the R2700 switching card (NM2-24ESW/NM2-16ESW) via SNMP, NM2-16ESW obtains the inexistent error message of port 17-26, while NM2-24ESW obtains the inexistent error message of port 25-26.

SNMP Security

Both SNMPv1 and SNMPv2 use the community string to check whether the management workstation is entitled to use MIB objects. In order to manage devices, the community string of NMS must be identical to a community string defined in the devices.

A community string features:

- Read-only: Authorized management workstations are entitled to read all the variables in the MIB.
- Read-write: Authorized management workstations are entitled to read and write all the variables in the MIB.

Based on SNMPv2, SNMPv3 can determine a security mechanism for processing data by security model and security level. There are three types of security models: SNMPv1, SNMPv2C and SNMPv3.

The table below describes the supported security models and security levels.

Model	Level	Authentication	Encryption	Description
SNMPv1	noAuthNoPriv	Community string	None	Ensures the data validity through community string.
SNMPv2c	noAuthNoPriv	Community string	None	Ensures the data validity through community string.
SNMPv3	noAuthNoPriv	User name	None	Ensures the data validity through user name.
SNMPv3	authNoPriv	MD5 or SHA	None	Provides HMAC-MD5 or HMAC-SHA-based authentication mechanism.
SNMPv3	authPriv	MD5 or SHA	DES	Provides HMAC-MD5 or HMAC-SHA-based authentication mechanism and CBC-DES-based encryption mechanism.

SNMP Engine ID

The engine ID is designed to identify a SNMP engine uniquely. Every SNMP entity contains a SNMP engine, a SNMP engine ID identifies a SNMP entity in a management domain. So every SNMPV3 entity has a unique identifier named SNMP Engine ID.

The SNMP Engine ID is an octet string of 5 to 32 bytes, which is defined in RFC3411:

- The first four bytes indicate the private enterprise number of an enterprise (assigned by IANA) in hex system.
- The fifth byte indicates how to identify the rest bytes.

0: Reserved

1: The following 4 bytes indicate an IPv4 address.

2: The following 16 bytes indicate an IPv6 address.

3: The following 6 bytes indicate an MAC address

4: Texts of up to 27 bytes defined by manufacturers

5: A hexadecimal value of up to 27 bytes defined by manufacturers

6-127: Reserved

128-255: In the format specified by manufacturers.

SNMP Configuration

The SNMP configuration is performed in global configuration mode on network devices. To configure SNMP, enter the global configuration mode.

Setting the Community String and Access Authority

SNMPv1 and SNMPv2C adopt community string-based security scheme. The SNMP Agent supports only the management operations from the management workstations of the same community string. The SNMP messages without matching the community string will be discarded. The community string serves as the password between the NMS and the SNMP Agent.

- Configure an ACL rule to allow the NMS of the specified IP address to manage devices.
- Set the community's operation permission,: ReadOnly or ReadWrite.
- Specify a view for view-based management. By default, no view is configured. That is, the management workstation is allowed to access to all MIB objects.
- Indicate the IP address of the NMS who can use this community string. If it is not indicated, any NMS can use this community string. By default, any NMS can use this community string.

To configure the SNMP community string, run the following command in global configuration mode:

	Command	Function
Step 1	Ruijie(config)# **snmp-server community** *string* [**view** *view-name*] [**ro** | **rw**] [**host** *host-ip*] [**ipv6** *ipv6-aclname*][*aclnum* | *aclname*]	Sets the community string and its right.

One or more coummnity strings can be specified for the NMS of different rights. To remove the community name and its right, run the **no snmp-server community** *string* command in global configuration mode.

Configuring MIB Views and Groups

With view-based access control model, you can determine whether the object of a management operation is in a view or not. Only the management objects in a view are allowed to be accessed. For access control, generally some users are associated with a group and then the group is associated with a view. The users in a group have the same access right.

- Set an inclusion view and an exclusion view.
- Set a Read-only view and a Read-write view for a group.
- Set security levels, whether to authenticate, and whether to encrypt for SNMPv3 users.

To configure the MIB views and groups, run the following commands in global configuration mode:

	Command	Function
Step 1	Ruijie(config)# **snmp-server view** *view-name* *oid-tree* **{include | exclude}**	Creates a MIB view to include or exclude associated MIB objects.

Step 2	Ruijie(config)# **snmp-server group** *groupname* {**v1 | v2c |v3 {auth | noauth | priv**}} [read *readview*] [**write** *writeview*] [**access** {[**ipv6** *ipv6_aclname*] [*aclnum* | *aclname*] }]	Creates a group and associate it with the view.

You can delete a view by using the **no snmp-server view** *view-name* command, or delete a tree from the view by using the **no snmp-server view** *view-name oid-tree* command. You can also delete a group by using the **no snmp-server group** *groupname* {**v1 \| v2c \| v3**} command.

Configuring SNMP Users

User-based security model can be used for security management. In this mode, you should configure user information first. The NMS can communicate with the SMP Agent by using a valid user account.

For SNMPv3 users, you can specify security level, authentication algorithm (MD5 or SHA), authentication password, encryption algorithm (only DES now) and encryption password.

To configure a SNMP user, run the following commands in global configuration mode:

Command	Function
Ruijie(config)# **snmp-server user** *username roupname* {**v1 | v2c | v3 [encrypted] [auth { md5|sha }** *auth-password*] **[priv des56** *priv-password*] } [**access {[ipv6** *ipv6_aclname*] [*aclnum* | *aclname*] }]	Configures the user information.

To remove the specified user, use the **no snmp-server user** *username groupname* {**v1 \| v2c \| v3**} command.

Configuring Host Address

In special cases, the SNMP Agent may also proactively send messages to the NMS.

To configure the NMS host address that the SNMP Agent proactively sends messages to, run the following commands in global configuration mode:

Command	Function
Ruijie(config)# **snmp-server host** { *host-addr* | ipv6 *ipv6-addr* } [**vrf** *vrfname*] [**traps**] [**version { 1 | 2c | 3 { auth | noauth | priv** }] *community-string* [**udp-port** *port-num*] [*notification-type*]	Sets the SNMP host address, host port, vrf options, message type, community string, (user name in SNMPv3) and security level (only supported in SNMPv3).

Configuring SNMP Agent Parameters

You can configure the basic parameters of the SNMP Agent, including contact, device location and sequence number. With these parameters, the NMS knows the contact, location and other information of the device.

To configure the SNMP agent parameters, run the following commands in global configuration mode:

Command	Function
Ruijie(config)# **snmp-server contact** *text*	Configures the contact for the system.
Ruijie(config)# **snmp-server location** *text*	Configure the location of the system.

Command	Function
Ruijie(config)# **snmp-server chassis-id** *number*	Configure the sequence number of the system.

Defining the Maximum Message Size of the SNMP Agent

In order to reduce influence on network bandwith, you can configure the maximum packet size of the SNMP Agent. To configure the maximum packet size of the SNMP Agent, run the following command in global configuration mode:

Command	Function
Ruijie(config)# **snmp-server packetsize** *byte-count*	Sets the maximum packet size of the SNMP Agent.

Shielding the SNMP Agent

The SNMP Agent service is a service provided by Ruijie product and it is enabled by default. You can shield the SNMP agent service and related configuration by executing the following command in global configuration mode:

Command	Function
Ruijie(config)# **no snmp-server**	Shields the SNMP agent service.

Disabling the SNMP Agent

Ruijie products provide a different command from the shield command to disable the SNMP Agent. This command will directly disable all SNMP services (the SNMP agent function is disabled, no message is received and no response or Trap message is sent) instead of shielding the configuration information of the SNMP Agent. To disable the SNMP agent service, run the following command in global configuration mode:

Command	Function
Ruijie(config)# **no enable service snmp-agent**	Disables the SNMP agent service.

Configuring the SNMP Agent to Send the Trap Message to the NMS Initiatively

The TRAP message is a message automatically sent by the SNMP Agent to the NMS, and is used to report some critical and important events. By default the SNMP Agent is not allowed to automatically send the TRAP message. To enable it, run the following command in global configuration mode:

Command	Function
Ruijie(config)# **snmp-server enable traps** [*type*] [*option*]	Allows the SNMP Agent to send the TRAP message proactively.
Ruijie(config)# **no snmp-server enable traps** [*type*] [*option*]	Forbids the SNMP Agent to send the TRAP message proactively.

Configuring LinkTrap Policy

You can configure whether to send the LinkTrap message on an interface. When this function is enabled and the link status of the interface changes, the SNMP will send the LinkTrap message. Otherwise, it will not. By default, this function is enabled.

Command	Function
Ruijie(config)# **interface** *interface-id*	Enters interface configuration mode.
Ruijie(config-if)# [**no**] **snmp trap link-status**	Enables or disables sending the LinkTrap message on the interface.

The following example configures not to send LinkTrap message on the intereface:

```
Ruijie(config)# interface gigabitEthernet 1/1
Ruijie(config-if)# no snmp trap link-status
```

Configuring the Parameters for Sending the Trap Message

To set the parameters for the SNMP Agent to send the Trap message, run the following commands:

Command	Function
Ruijie(config)# **snmp-server trap-source** *interface*	Specifies the source port sending the Trap message.
Ruijie(config)# **snmp-server queue-length** *length*	Specifies the queue length of each Trap message.
Ruijie(config)# **snmp-server trap-timeout** *seconds*	Specifies the interval at which the Trap message is sent.

Configuring SNMP Attack Protection

Enable SNMP attack protection by confining limited times of failed SNMP consecutive authentications and specifying the solution after consecutive authentications fail. After SNMP authentications fail, the system will blacklist the source IP. When the failed times exceed the limit, the system will restrict the source IP address according to the solutions configured by the device:

- The source IP address that is prevented from authentication permanently cannot try access authentication again unless it is relieved by the administrator manually.
- The source IP address that is prevented from authentication for a while can try access authentication again when the **lock-time** times out or it is relieved by the administrator manually.
- When you try access authentication again, the non-restricted source IP address will pass it as long as you use correct community (for SNMPv1and SNMPv2c) or username (for SNMPv3)..

Run this command in global configuration mode to limit the times of failed SNMP consecutive authentications and specify the solution after consecutive authentications fail.

Command	Function

Command	Function
Ruijie(config)# **snmp-server authentication attempt** *times* **exceed** { **lock** | **lock-time** *minutes* | **unlock** }	Sets limited times of failed SNMP consecutive authentications and specifies the solution after authentications fail. **attempt** *times*: The limit of failed SNMP authentications. **Lock**: The source IP address is prevented from access authentication permanently. It is blacklisted unless relieved by the administrator manually. **lock-time** *minutes*: The source IP address is prevented from access authentication for a while and then allowed to be authenticated again. *minutes* refers to the period when the source IP address is prevented. **unlock**: The source IP address continues to be allowed after consecutive authentications fail, similar to the case that SNMP attack protection is not enabled.

Run the **no snmp-server authentication attempt** command to restore SNMP attack protection. By default, the solution taken after consecutive authentications fail is **unlock**. Namely, the IP address is allowed to try access authentication.

SNMP Monitoring and Maintenance

Checking the Current SNMP Status

To monitor the SNMP status and troubleshoot SNMP configurations, Ruijie product provides monitoring commands for SNMP, with which it is possible to easily check the SNMP status of the current network device. In privileged user mode, run the **show snmp** command to check the current SNMP status.

```
Ruijie# show snmp
Chassis: 1234567890 0987654321
Contact: wugb@i-net.com.cn
Location: fuzhou
2381 SNMP packets input
    5 Bad SNMP version errors
    6 Unknown community name
    0 Illegal operation for community name supplied
    0 Encoding errors
    9325 Number of requested variables
    0 Number of altered variables
    31 Get-request PDUs
    2339 Get-next PDUs
```

```
   0 Set-request PDUs
2406 SNMP packets output
   0 Too big errors (Maximum packet size 1500)
   4 No such name errors
   0 Bad values errors
   0 General errors
   2370 Get-response PDUs
   36 SNMP trap PDUs
SNMP global trap: disabled
SNMP logging: enabled
SNMP agent: enabled
.
```

The above statistics is explained as follows:

Showing Information	Description
Bad SNMP version errors	SNMP version is incorrect.
Unknown community name	The community name cannot be identified.
Illegal operation for community name supplied	Illegal operation
Encoding errors	Code error
Get-request PDUs	Get-request message
Get-next PDUs	Get-next message
Set-request PDUs	Set-request message
Too big errors (Maximum packet size 1500)	Too large response message
No such name errors	The specified management unit does not exist.
Bad values errors	Specified value type error
General errors	General error
Get-response PDUs	Get-response message
SNMP trap PDUs	SNMP trap message

Checking the MIB Objects Supported by the Current SNMP Agent

To check the MIB objects supported by the current SNMP Agent, run the **show snmp mib** command in privileged user mode:

```
Ruijie# show snmp mib
sysDescr
sysObjectID
sysUpTime
sysContact
sysName
sysLocation
sysServices
sysORLastChange
snmpInPkts
snmpOutPkts
```

```
...
```

Viewing SNMP Users

To view the SNMP users configured on the current SNMP agent, run the **show snmp user** command in privileged user mode:

```
Ruijie# show snmp user
User name: test
Engine ID: 8000131103000000000000
storage-type: permanent    active
Security level: auth priv
Auth protocol: SHA
Priv protocol: DES
Group-name: g1
```

Viewing SNMP Views and Groups

To view the group configured on the current SNMP agent, run the **show snmp group** command in privileged user mode:

```
Ruijie# show snmp group
groupname: g1
securityModel: v3
securityLevel:authPriv
readview: default
writeview: default
notifyview:
groupname: public
securityModel: v1
securityLevel:noAuthNoPriv
readview: default
writeview: default
notifyview:
groupname: public
securityModel: v2c
securityLevel:noAuthNoPriv
readview: default
writeview: default
notifyview:
```

To view the view configured on the current SNMP agent, run the **show snmp view** command in privileged user mode:

```
Ruijie# show snmp view
default(include) 1.3.6.1
test-view(include) 1.3.6.1.2.1
```

Viewing Host Information

To view the host information configured on the SNMP agent, run the **show snmp host** command in privileged user mode:

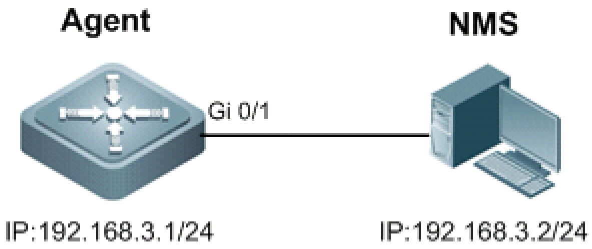
```
Ruijie# show snmp host
Notification host: 192.168.64.221
udp-port: 162   type: trap
user: public    security model: v1
Notification host: 2000:1234::64
udp-port: 162   type: trap
user: public    security model: v1
```

Typical SNMP Configuration Example

SNMP v1/v2 Configuration Example

Topological Diagram

Figure 4 Topology for SNMP v1/2 application



Application Requirements

The Network Management Station (NMS) manages the network device (Agent) by applying the community-based authentication model, and the network device can control the operation permission (read or write) of the community to access the specified MIB objects. For example, community "user1" can only read and write objects under System (1.3.6.1.2.1.1) node.

The network device can only be managed by NMS with a specific IP (i.e., 192.168.3.2/24).

The network device can actively send messages to NMS.

The NMS can acquire the basic system information of the device, such as contact, location, and ID.

Configuration Tips

By creating MIB view and associating authentication name (Community) and access permission (Read or Write), the first application need can be met.

While configuring the community string and access permission, associate ACL or specify the IP of administrator using this community string to meet the second application need (this example associates the ACL).

Configure the address of SNMP host and enable the Agent to actively send Trap messages.

Configure the parameters of SNMP agent.

Configuration Steps

Step 1: Configure MIB view and ACL.

! Create an MID view named "v1", which contains the associated MIB object (1.3.6.1.2.1.1).

```
Ruijie(config)#snmp-server view v1 1.3.6.1.2.1.1 include
```

! Create an ACL named "a1" to permit the IP address of 192.168.3.2/24.

```
Ruijie(config)#ip access-list standard a1
Ruijie(config-std-nacl)#permit host 192.168.3.2
Ruijie(config-std-nacl)#exit
```

Step 2: Configure community string and access permission.

! Configure Community of "user1", associate write permission for MIB view of "v1", and associate the ACL of "a1".

```
Ruijie(config)#snmp-server community user1 view v1 rw a1
```

Step 3: Configure the Agent to actively send messages to NMS.

! Configure the address of SNMP host as 192.168.3.2, message format as Version 2c and community string as "user1".

```
Ruijie(config)#snmp-server host 192.168.3.2 traps version 2c user1
```

! Enable the Agent to actively send traps.

```
Ruijie(config)#snmp-server enable traps
```

Step 4: Configure parameters of SNMP agent.

! Configure system location.

```
Ruijie(config)#snmp-server location fuzhou
```

! Configure system contact.

```
Ruijie(config)#snmp-server contact ruijie.com.cn
```

! Configure system serial number.

```
Ruijie(config)#snmp-server chassis-id 1234567890
```

Step 5: Configure the IP address for the Agent.

! Configure the IP address of Gi 0/1 as 192.168.3.1/24.

```
Ruijie(config)#interface GigabitEthernet 0/1
Ruijie(config-if-GigabitEthernet 0/1)#ip address 192.168.3.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 0/1)#exit
```

Verification

Step 1: Display configurations of the device.

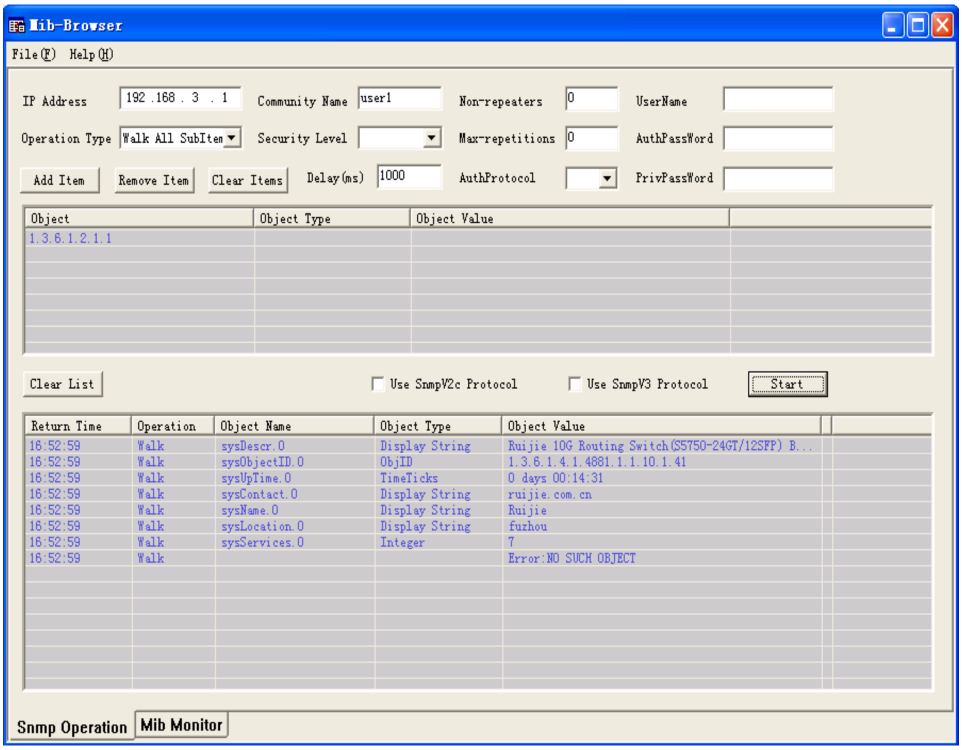
```
Ruijie#show running-config
!
ip access-list standard a1
 10 permit host 192.168.3.2
!
interface GigabitEthernet 0/1
 no ip proxy-arp
 ip address 192.168.3.1 255.255.255.0
!
snmp-server view v1 1.3.6.1.2.1.1 include
snmp-server location fuzhou
snmp-server host 192.168.3.2 traps version 2c user1
snmp-server enable traps
snmp-server contact ruijie.com.cn
snmp-server community user1 view v1 rw a1
snmp-server chassis-id 1234567890
```

Step 2: Display information about SNMP view and group.

```
Ruijie#show snmp view
v1(include) 1.3.6.1.2.1.1                //define MIB object of “v1”
default(include) 1.3.6.1                 //default MIB object
Ruijie#show snmp group
groupname: user1                  //Configure Community as SNMP group
securityModel: v1
securityLevel:noAuthNoPriv
readview: v1
writeview: v1
notifyview:
groupname: user1
securityModel: v2c
securityLevel:noAuthNoPriv
readview: v1
writeview: v1
notifyview:
```

Step 3: Install MIB-Browser. Type in device IP of "192.168.3.1" in the field of IP Address; type in "user1" in the field of Community Name; click **Add Item** button and select the specific management unit for MIB query, such as the System shown below. Click **Start** to implement MIB query of network device. The query result is shown in the bottommost box:

Figure 5



IP Address 192 .168 . 3 . 1

Community Name user1

Non-repeaters 0

UserName

Operation Type Walk All SubIten

Security Level

Max-repetitions 0

AuthPassWord

Delay(ms) 1000

AuthProtocol

PrivPassWord

Object	Object Type	Object Value
1.3.6.1.2.1.1		

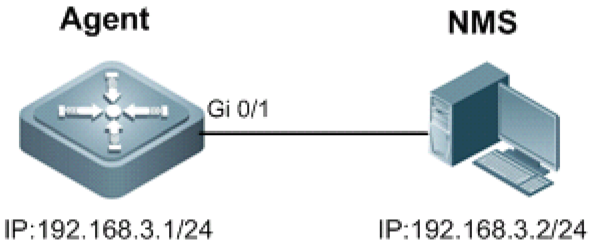
□ Use SnmpV2c Protocol □ Use SnmpV3 Protocol

Return Time	Operation	Object Name	Object Type	Object Value
16:52:59	Walk	sysDescr.0	Display String	Ruijie 10G Routing Switch(S5750-24GT/12SFP) B...
16:52:59	Walk	sysObjectID.0	ObjID	1.3.6.1.4.1.4881.1.1.10.1.41
16:52:59	Walk	sysUpTime.0	TimeTicks	0 days 00:14:31
16:52:59	Walk	sysContact.0	Display String	ruijie.com.cn
16:52:59	Walk	sysName.0	Display String	Ruijie
16:52:59	Walk	sysLocation.0	Display String	fuzhou
16:52:59	Walk	sysServices.0	Integer	7
16:52:59	Walk			Error:NO SUCH OBJECT

SNMP v3 Configuration Example

Topological Diagram

Figure 6 SNMPv3 application topology



Application Requirements

Network Management Station manages the network device (Agent) by applying user-based security model. For example: the user name is "user1", authentication mode is MD5, authentication key is "123", encryption algorithm is DES56, and the encryption key is "321".

The network device can control user's permission to access MIB objects. For example: "User1" can read the MIB objects under System (1.3.6.1.2.1.1) node, and can only write MIB objects under SysContact (1.3.6.1.2.1.1.4.0) node.

The network device can actively send authentication and encryption messages to the network management station.

Configuration Tips

Create MIB view and specify the included or excluded MIB objects.

Create SNMP group and configure the version as "v3"; specify the security level of this group, and configure the read/write permission of the view corresponding to this group.

Create user name and associate the corresponding SNMP group name to further configure the user's permission to access MIB objects; meanwhile, configure the version number as "v3" and the corresponding authentication mode, authentication key, encryption algorithm and encryption key.

Configure the address of SNMP host, configure the version number as "3" and configure the security level to be adopted.

Configuration Steps

Step 1: Configure MIB view and group.

! Create an MIB view of "view1" and include the MIB object of 1.3.6.1.2.1.1; further create an MIB view of "view2" and include the MIB object of 1.3.6.1.2.1.1.4.0.

```
Ruijie(config)#snmp-server view view1 1.3.6.1.2.1.1 include
Ruijie(config)#snmp-server view view2 1.3.6.1.2.1.1.4.0 include
```

! Create a group named "g1" and select the version number of "v3"; configure security level to "priv" to read "view1" and write "view2".

```
Ruijie(config)#snmp-server group g1 v3 priv read view1 write view2
```

Step 2: Configure SNMP user.

! Create a user named "user1", which belongs to group "g1"; select version number of "v3" and configure authentication mode as "md5", authentication key as "123", encryption mode as "DES56" and encryption key as "321".

```
Ruijie(config)#snmp-server user user1 g1 v3 auth md5 123 priv des56 321
```

Step 3: Configure the address of SNMP host.

! Configure the host address as 192.168.3.2 and select version number of "3"; configure security level to "priv" and associate the corresponding user name of "user1".

```
Ruijie(config)#snmp-server host 192.168.3.2 traps version 3 priv user1
```

! Enable the Agent to actively send traps to NMS.

```
Ruijie(config)#snmp-server enable traps
```

Step 4: Configure the IP address of Agent.

! Configure the IP address of Gi 0/1 as 192.168.3.1/24.

```
Ruijie(config)#interface gigabitEthernet 0/1
Ruijie(config-if-GigabitEthernet 0/1)#ip address 192.168.3.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 0/1)#exit
```

Verification

Step 1: Display configurations of device.

```
Ruijie#show running-config
```

```
!
interface GigabitEthernet 0/1
 no ip proxy-arp
 ip address 192.168.3.1 255.255.255.0
!
snmp-server view view1 1.3.6.1.2.1.1 include
snmp-server view view2 1.3.6.1.2.1.1.4.0 include
snmp-server user user1 g1 v3 encrypted auth md5 7EBD6A1287D3548E4E52CF8349CBC93D priv des56
D5CEC4884360373ABBF30AB170E42D03
snmp-server group g1 v3 priv read view1 write view2
snmp-server host 192.168.3.2 traps version 3 priv user1
snmp-server enable traps
```

Step 2: Display SNMP user.

```
Ruijie# show snmp user
User name: user1
Engine ID: 800013110300d0f8221120
storage-type: permanent    active
Security level: auth priv
Auth protocol: MD5
Priv protocol: DES
Group-name: g1
```

Step 3: Display SNMP view.

```
Ruijie#show snmp view
view1(include) 1.3.6.1.2.1.1
view2(include) 1.3.6.1.2.1.1.4.0
default(include) 1.3.6.1
```

Step 4: Display SNMP group.

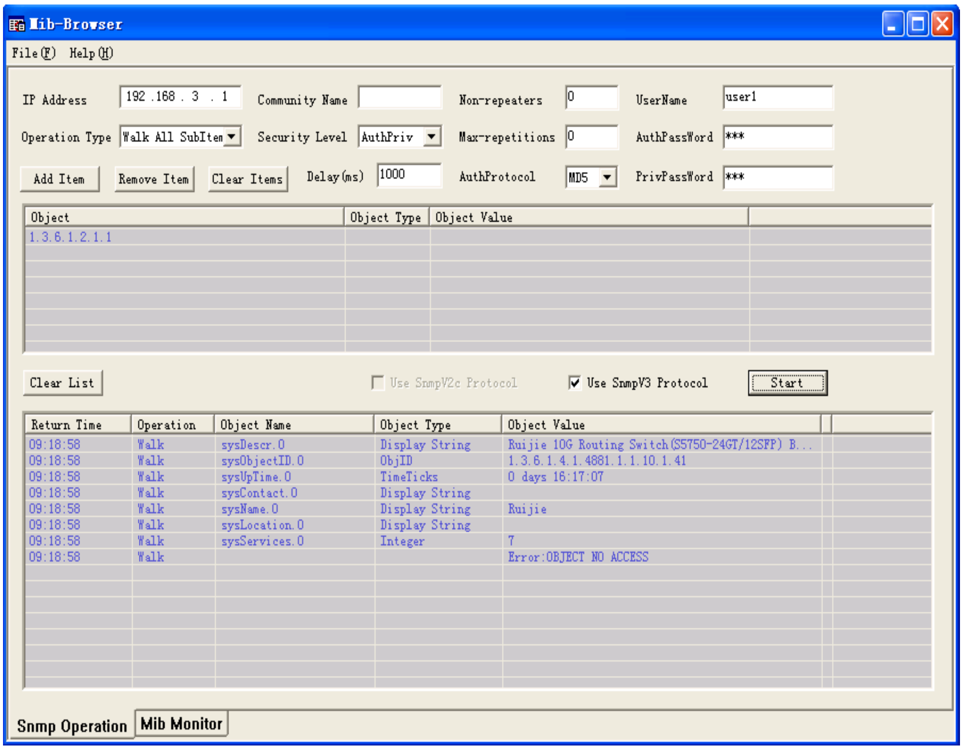
```
Ruijie# show snmp group
groupname: g1
securityModel: v3
securityLevel:authPriv
readview: view1
writeview: view2
notifyview:
```

Step 5: Display host information configured by the user.

```
Ruijie#show snmp host
Notification host: 192.168.3.2
udp-port: 162
type: trap
user: user1
security model: v3 authPriv
```

Step 6: Install MIB-Browser. Type in device IP of "192.168.3.1" in the field of IP Address; type in "user1" in the field of UserName; select "AuthPriv" from Security Level; type in "123" in the field of AuthPassWord; select "MD5" from AuthProtocol; type in "321" in the field of PrivPassWord. Click **Add Item** button and select the specific management unit for querying MIB, such as the System shown below. Click **Start** to implement MIB query of network device. The query result is shown in the bottommost box:

Figure 7



IP Address	192 .168 . 3 . 1	Community Name		Non-repeaters	0	UserName	user1
Operation Type	Walk All SubItem	Security Level	AuthPriv	Max-repetitions	0	AuthPassWord	***
		Delay(ms)	1000	AuthProtocol	MD5	PrivPassWord	***

Object	Object Type	Object Value
1.3.6.1.2.1.1		

□ Use SnmpV2c Protocol ☑ Use SnmpV3 Protocol

Return Time	Operation	Object Name	Object Type	Object Value
09:18:58	Walk	sysDescr.0	Display String	Ruijie 10G Routing Switch(S5750-24GT/12SFP) B...
09:18:58	Walk	sysObjectID.0	ObjID	1.3.6.1.4.1.4881.1.1.10.1.41
09:18:58	Walk	sysUpTime.0	TimeTicks	0 days 16:17:07
09:18:58	Walk	sysContact.0	Display String	
09:18:58	Walk	sysName.0	Display String	Ruijie
09:18:58	Walk	sysLocation.0	Display String	
09:18:58	Walk	sysServices.0	Integer	7
09:18:58	Walk			Error:OBJECT NO ACCESS

Configuring CWMP

Overview

CPE WAN Management Protocol (CWMP) provides a general framework of unified device management, related message specifications, management methods, and data models, so as to solve difficulties in unified management and maintenance of dispersed customer-premises equipment (CPEs), improve troubleshooting efficiency, and save O&M costs.

CWMP provides the following functions:

- **Auto configuration and dynamic service provisioning.** CWMP allows an Auto-Configuration Server (ACS) to automatically provision CPEs who initially access the network after start. The ACS can also dynamically re-configure running CPEs.
- **Firmware management.** CWMP manages and upgrades the firmware and its files of CPEs.
- **Software module management.** CWMP manages modular software according to data models implemented.
- **Status and performance monitoring.** CWMP enables CPEs to notify the ACE of its status and changes, achieving real-time status and performance monitoring.
- **Diagnostics.** The ACE diagnoses or resolves connectivity or service problems based on information from CPEs, and can also perform defined diagnosis tests.

Protocols and Standards

For details about TR069 protocol specifications, visit http://www.broadband-forum.org/technical/trlist.php.

Listed below are some major CWMP protocol specifications:

- TR-069_Amendment-4.pdf: CWMP standard
- TR-098_Amendment-2.pdf: Standard for Internet gateway device data model
- TR-106_Amendment-6.pdf: Standard for CPE data model
- TR-181_Issue-2_Amendment-5.pdf: Standard for CPE data model 2
- tr-098-1-4-full.xml: Definition of Internet gateway device data model
- tr-181-2-4-full.xml: Definition 2 of CPE data model 2

Applications

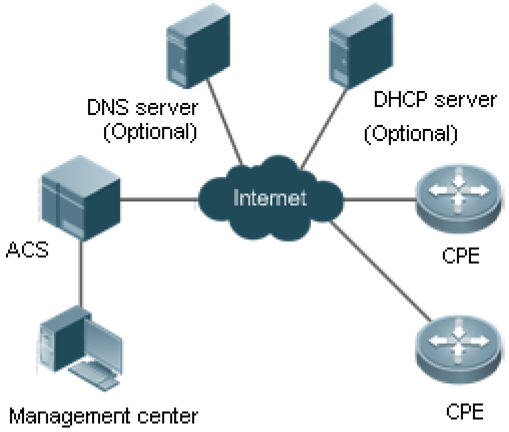
Typical Application	Scenario
CWMP Network Application Scenario	Initiate CPE-ACS connection, so as to upgrade the CPE firmware, upload the configuration files, restore the configuration, and realize other features.

CWMP Network Application Scenario

Application Scenario

The major components of a CWMP network architecture are CPEs, an ACS, a management center, a DHCP server, and a Domain Name System (DNS) server. The management center manages a population of CPEs by controlling the ACS on a Web browser.

Figure 0-1



Note

- If the Uniform Resource Locator (URL) of the ACS is configured on CPEs, the DHCP server is optional. If not, the DHCP is required to dynamically discover the ACS URL.
- If the URLs of the ACS and CPEs contain IP addresses only, the DNS server is optional. If their URLs contain domain names, the DNS server is required to resolves the names.

Functional Deployment

HTTP runs on both CPEs and the ACS.

Features

Basic Concept

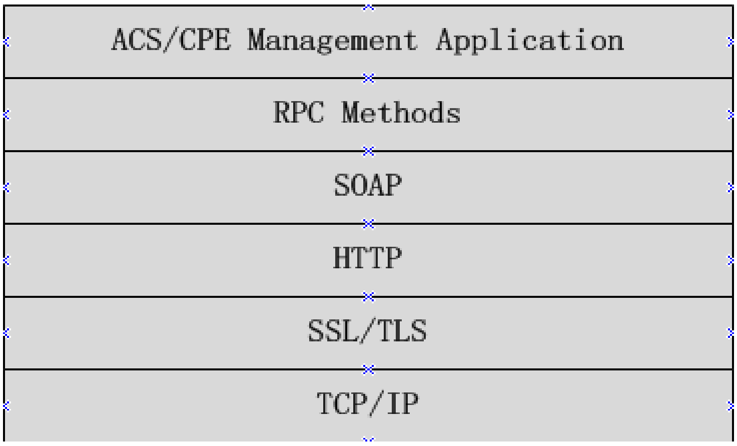
↘ Major Terminologies

- **CPE**: Customer Premises Equipment
- **ACS**: Auto-Configuration Server
- **RPC**: Remote Procedure Call
- **DM**: Data Model

↘ Protocol Stack

Figure 0-2 shows the protocol stack of CWMP.

Figure 0-2 CWMP Protocol Stack



As shown in 错误!未找到引用源。, CWMP defines six layers with respective functions as follows:

- **ACS/CPE Application**

The application layer is not a part of CWMP. It is the development performed by various modules of the CPEs/ACS to support CWMP, just like the Simple Network Management Protocol (SNMP), which does not cover the MIB management of functional modules.

- **RPC Methods**

This layer provides various RPC methods for interactions between the ACS and the CPEs.

- **SOAP**

The Simple Object Access Protocol (SOAP) layer uses a XML-based syntax to encode and decode CWMP messages.. Thus, CWMP messages must comply with the XML-based syntax.

- **HTTP**

All CWMP messages are transmitted over Hypertext Transfer Protocol (HTTP). Both the ACS and the CPEs can behave in the role of HTTP clients and servers. The server function is used to monitor reverse connections from the peer.

- **SSL/TLS**

The Secure Sockets Layer (SSL) or Transport Layer Security (TLS) layer guarantees CWMP security, including data integrity, confidentiality, and authentication.

- **TCP/IP**

This layer is the (Transmission Control Protocol/Internet Protocol (TCP/IP) protocol stack.

RPC Methods

The ACS manages and monitors CPEs by calling mostly the following RPC methods:

- **Get RPC Methods**

The Get methods enable the ACS to remotely obtain the set of RPC methods, as well as names, values and attributes of the DM parameters supported on CPEs.

- **Set RPC Methods**

The Set methods enable the ACS to remotely set the values and attributes of the DM parameters supported on CPEs.

- **Inform RPC Methods**

The Inform methods enable CPEs to inform the ACS of their device identifiers, parameter information, and events whenever sessions are established between them.

- **Download RPC Methods**

The Download method enables the ACS to remotely control the file download of CPEs, including firmware management, upgrade, and Web package upgrade.

- **Upload RPC Methods**

The Upload method enables the ACS to remotely control the file upload of CPEs, including upload of firmware and logs.

- **Reboot RPC Methods**

The Reboot method enables the ACS to remotely reboot the CPEs.

↘ Session Management

CWMP sessions or interactions are the basis for CWMP. All CWMP interactions between the ACS and CPEs rely on their sessions. CWMP helps initiate and maintain ACS-CPE sessions to link them up for effective management and monitoring. An ACS-CPE session is a TCP connection, which starts from the Inform negotiation to TCP disconnection. The session is classified into CPE Initiated Session and ACS Initiated Session according to the session poster.

↘ DM Management

CWMP operates based on CWMP Data Model (DM). CWMP manages all functional modules by a set of operations performed on DM. Each functional module registers and implements a respective data model, just like the MIBs implemented by various functional modules of SNMP.

A CWMP data model is represented in the form of a character string. For a clear hierarchy of the data model, a dot (.) is used as a delimiter to distinguish an upper-level data model node from a lower-level data model node. For instance, in the data model **InternetGatewayDevice.LANDevice**, **InternetGatewayDevice** is the parent data model node of **LANDevice**, and **LANDevice** is the child data model node of **InternetGatewayDevice**.

DM nodes are classified into two types: object nodes and parameter nodes. The parameter nodes are also known as leaf nodes. An object node is a node under which there are child nodes, and a parameter node is a leaf node under which there is no any child node. Object nodes are further classified into single-instance object nodes and multi-instance object nodes. A single-instance object node is an object node for which there is only one instance, whereas a multi-instance object node is an object node for which there are multiple instances.

DM nodes can also be classified into readable nodes and readable-and-writable nodes. A readable node is a node whose parameter values can be read but cannot be modified, and a readable-and-writable node is a node whose parameter values can be both read and modified.

A data model node has two attributes. One attribute relates to a notification function; that is, whether to inform the ACS of changes (other than changes caused by CWMP) to parameter values of the data model. The other attribute is an identifier indicating that the parameters of the data model node can be written using other management modes (than the ACS); that is, whether the values of the parameters can be modified using other management modes such as Telnet. The ACS can modify the attributes of the data models using RPC methods.

CWMP manages the data models using corresponding RPC methods.

↘ Event Management

When some events concerned by the ACS occur on the CPE, the CPE will inform the ACS of these events. The ACS monitors these events to monitor the working status of the CPE. The CWMP events are just like Trap messages of SNMP or product logs. Using RPC methods, to the ACS filters out the unconcerned types of events. CWMP events are classified into two types: single or (not cumulative) events and multiple (cumulative) events. A single event means that there is no quantitative change to the same event upon re-occurrence of the event, with the old discarded and the newest kept. A multiple event means that the old are not discarded and the newest event is kept as a complete event when an event re-occurs for multiple times later; that is, the number of this event is incremented by 1.

All events that occur on the CPE are notified to the ACS using the INFORM method.

Features

Feature	Description
Upgrading the Firmware	The ACS controls the upgrade of the firmware of a CPE using the Download method.
Upgrading the Configuration Files	The ACS controls the upgrade of the configuration files of a CPE using the Download method.
Uploading the Configuration Files	The ACS controls the upload of the configuration files of a CPE using the Upload method.
Backing up and Restoring a CPE	When a CPE breaks away from the management center, this feature can remotely restore the CPE to the previous status.

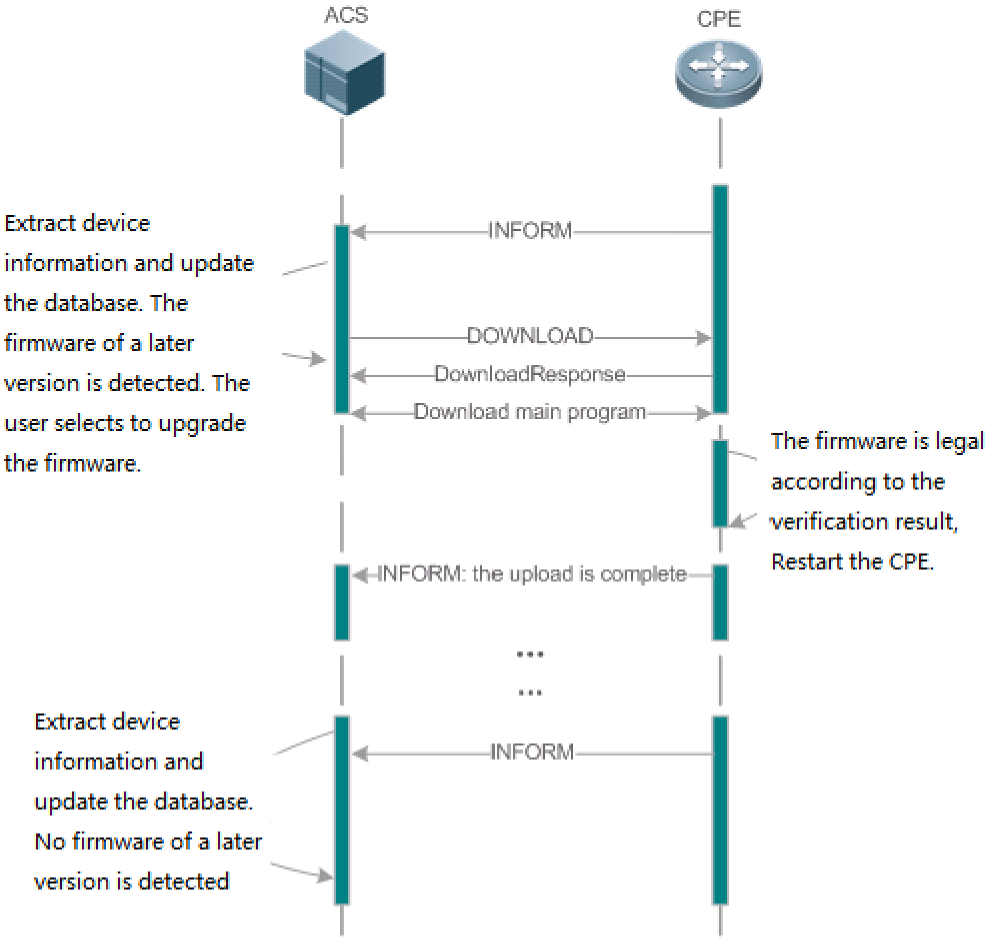
Upgrading the Firmware

Upgrading the Firmware means the firmware of a network element (NE) can be upgraded, so as to implement device version upgrade or replacement.

Working Principle

↘ Sequence Diagram of Upgrading the Firmware

Figure 0-3


ACS
CPE
Extract device information and update the database. The firmware of a later version is detected. The user selects to upgrade the firmware.
INFORM
DOWNLOAD
DownloadResponse
Download main program
The firmware is legal according to the verification result, Restart the CPE.
INFORM: the upload is complete
...
...
Extract device information and update the database. No firmware of a later version is detected
INFORM

Users specify a CPE for the ACS to deliver the Download method for upgrading the firmware. The CPE receives the request and starts to download the latest firmware from the destination file server, upgrade the firmware, and then reboot. After restart, the CPE will indicate the successful or unsuccessful completion of the method application.

Note

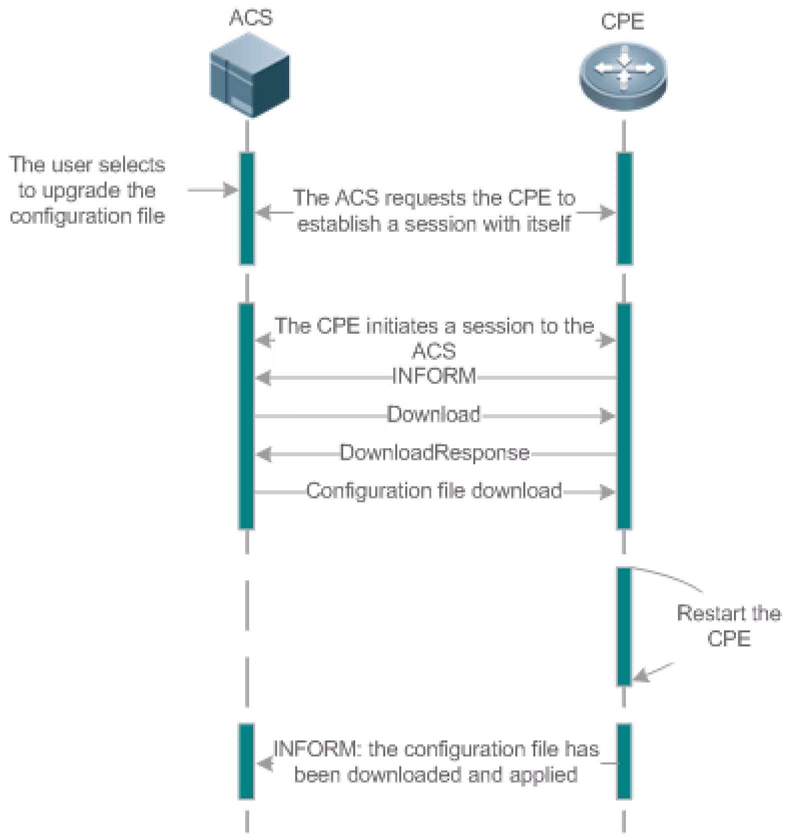
The file server can be ACS or separately deployed.

Upgrading the Configuration Files

Upgrading the Configuration Files means the current configuration files of a CPE can be replaced with specified configuration files, so that the new configuration files act on the CPE after reset.

Working Principle

Figure 0-4


ACS
CPE
The user selects to upgrade the configuration file
The ACS requests the CPE to establish a session with itself
The CPE initiates a session to the ACS
INFORM
Download
DownloadResponse
Configuration file download
Restart the CPE
INFORM: the configuration file has been downloaded and applied

Users specify a CPE for the ACS to deliver the Download methods for upgrading its configuration files. The CPE downloads the configuration files from the specified file server, upgrade configuration files, and then reboot. After that, the CPE will indicate successful or unsuccessful completion of the method application.

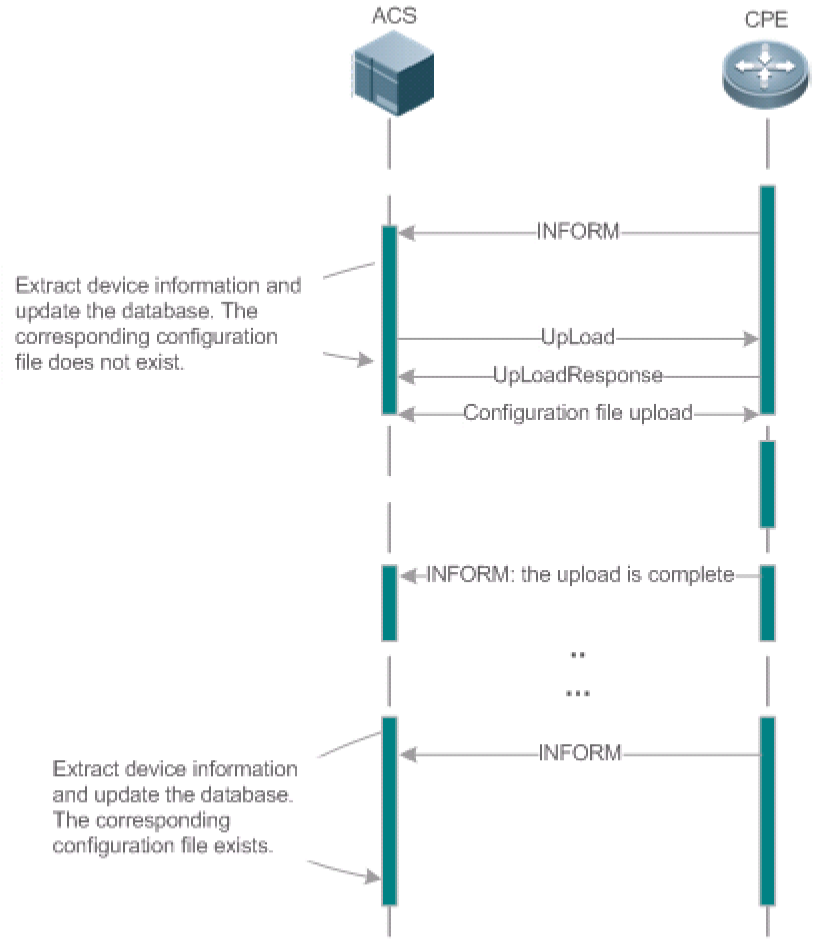
Note The file server can be ACS or separately deployed.

Uploading the Configuration Files

Uploading the Configuration Files means the ACS controls the configuration files of CPEs by using the Upload method.

Working Principle

Figure 0-5


ACS
CPE
INFORM
Extract device information and update the database. The corresponding configuration file does not exist.
UpLoad
UpLoadResponse
Configuration file upload
INFORM: the upload is complete
...
INFORM
Extract device information and update the database. The corresponding configuration file exists.

When a CPE initially accesses the ACS, the ACS attempts to learn the configuration files of the CPE in the following sequence:

- When the ACS initially receives an Inform message from the CPE, it locates the corresponding database information according to device information carried in the message.
- If the database does not contain the configuration files of the CPE, the ACS delivers the Upload method to the CPE for uploading the configuration files.
- The CPE uploads its current configuration files to the ACS.
- The CPE returns a successful or unsuccessful response to the Upload request.

Backing Up and Restoring a CPE

When a remote CPE breaks away from the management center due to abnormal operations, the CPE backup and restoration feature helps restore the CPE to the previous status, so that the management center can resume the supervision of the CPE as necessary.

Working Principle

You can configure the restoration function on a CPE, so that the CPE can restore itself from exceptions of its firmware or configuration files. Then when the CPE fails to connect to the ACS and breaks away from the management center after its firmware or configuration files are upgraded, the previous firmware or configuration files of the CPE can be restored in time for the ACS to manage the CPE. This kind of exception is generally caused by delivery of a wrong version or configuration file.

Before the CPE receives a new firmware or configuration files to upgrade, the CPE will back up its current version and configuration files. In addition, there is a mechanism for determining whether the problem described in the preceding scenario has occurred. If the problem has occurred, the CPE is restored to the previous manageable status.

Configuration

Action	Suggestions and Related Commands	
Establishing a Basic CWMP Connection	⚠ (Mandatory) You can configure the ACS or CPE usernames and passwords to be authenticated for CWMP connection.	
	cwmp	Enables CWMP and enters CWMP configuration mode.
	acs username	Configures the ACS username for CWMP connection.
	acs password	Configures the ACS password for CWMP connection.
	cpe username	Configures the CPE username for CWMP connection.
	cpe password	Configures the CPE password for CWMP connection.
	⚠ (Optional) You can configure the URLs of the CPE and the ACS.	
	acs url	Configures the ACS URL.
	cpe url	Configures the CPE URL.
Configuring CWMP-Related Attributes	⚠ (Optional) You can configure the basic functions of the CPE, such as upload, backup and restoration of firmware, configuration files or logs.	
	cpe inform	Configures the periodic notification function of the CPE.
	cpe back-up	Configures the backup and restoration of the firmware and configuration file of the CPE.
	disable download	Disables the function of downloading firmware and configuration files from the ACS.
	disable upload	Disables the function of uploading configuration and log files to the ACS.

Action	Suggestions and Related Commands	
	timer cpe- timeout	Configures the ACS response timeout on CPEs.

Establishing a Basic CWMP Connection

Configuration Effect

- A session connection is established between the ACS and the CPE.

Precautions

- N/A

Configuration Method

Enabling CWMP and Entering CWMP Configuration Mode

- (Mandatory) The CWMP function is enabled by default.

Command	**cwmp**
Parameter Description	N/A
Defaults	CWMP is enabled by default.
Command Mode	Global configuration guide
Usage Guide	N/A

Configuring the ACS Username for CWMP Connection

- This configuration is mandatory on the ACS.
- Only one username can be configured for the ACS. If multiple are configured, the latest configuration is applied.

Command	**acs username** *username*
Parameter Description	**username** *username*: The ACS username for CWMP connection
Defaults	The ACS username is not configured by default.
Command Mode	CWMP configuration mode
Usage Guide	N/A

Configuring the ACS Password for CWMP Connection

- This configuration is mandatory on the ACS.
- The password of the ACS can be in plaintext or encrypted form. Only one password can be configured for the ACS. If multiple are configured, the latest configuration is applied.

Command	**acs password** {*password* | *encryption-type encrypted-password*}
Parameter Description	*password:* ACS password *encryption-type:* 0 (no encryption) or 7 (simple encryption) *encrypted-password:* Password text
Defaults	*encryption-type*: 0

	encrypted-password: N/A
Command Mode	CWMP configuration mode
Usage Guide	N/A

Configuring the CPE Username for CWMP Connection

- This configuration is mandatory on the CPE.
- Only one username can be configured for the CPE. If multiple are configured, the latest configuration is applied.

Command	**cpe username** *username*
Parameter Description	*username*: CPE username
Defaults	No CPE username is configured by default.
Command Mode	CWMP configuration mode
Usage Guide	N/A

Configuring the CPE Password for CWMP Connection

- This configuration is mandatory on the CPE.
- The password of the CPE can be in plaintext or encrypted form. Only one password can be configured for the CPE. If multiple are configured, the latest configuration is applied.

Command	**cpe password** {*password* | *encryption-type encrypted-password*}
Parameter Description	*password:* CPE password *encryption-type:* 0 (no encryption) or 7 (simple encryption) *encrypted-password:* Password text
Defaults	*encryption-type*: 0 *encrypted-password*: N/A
Command Mode	CWMP configuration mode
Usage Guide	Use this command to configure the CPE user password to be authenticated for the ACS to connect to the CPE. In general, the encryption type does not need to be specified. The encryption type needs to be specified only when copying and pasting the encrypted password of this command. A valid password should meet the following format requirements: ● Contain 1 to 26 characters including letters and figures. ● The leading spaces will be ignored, while the trailing and middle are valid. ● If 7 (simple encryption) is specified, the valid characters only include 0 to 9 and a (A) to f (F).

Configuring the ACS URL for CMWP Connection

- This configuration is optional on the CPE.
- Only one ACS URL can be configured. If multiple are configured, the latest configuration is applied. The ACS URL must be in HTTP format.

Command	**acs url** *url*
Parameter Description	*url*: ACS URL
Defaults	No ACS URL is configured by default.

Command Mode	CWMP configuration mode
Usage Guide	If the ACS URL is not configured but obtained through DHCP, CPEs will use this dynamic URL to initiate connection to the ACS. The ACS URL must: ● Be in format of **http://host[:port]/path**. ● Contain 256 characters at most.

Configuring the CPE URL for CWMP Connection

- This configuration is optional on the CPE.
- Only one CPE URL can be configured. If multiple are configured, the latest configuration is applied. The CPE URL must be in HTTP format instead of domain name format.

Command	**cpe url** *url*
Parameter Description	*url:* CPE URL
Defaults	No CPE URL is configured by default.
Command Mode	CWMP configuration mode
Usage Guide	If CPE URL is not configured, it is obtained through DHCP. The CPE URL must: ● Be in format of **http://ip [: port]/**. ● Contain 256 characters at most.

Verification

- Run the **show cwmp configuration** command.

Command	**show cwmp configuration**
Parameter Description	N/A
Command Mode	Privileged EXEC mode
Usage Guide	N/A
Configuration Examples	The following example displays the CWMP configuration.

```
Ruijie(config-cwmp)#show cwmp configuration
CWMP Status                   : enable
ACS URL                      : http://www.ruijie.com.cn/acs
ACS username                  : admin
ACS password                  : ******
CPE URL                      : http://10.10.10.2:7547/
CPE username                  : ruijie
CPE password                  : ******
CPE inform status              : disable
CPE inform interval            : 60s
CPE inform start time          : 0:0:0 0 0 0
CPE wait timeout              : 50s
CPE download status            : enable
CPE upload status              : enable
CPE back up status             : enable
CPE back up delay time          : 60s
```

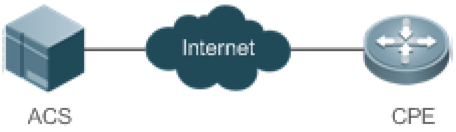
Configuration Examples

The following configuration examples describe CWMP-related configuration only.

Configuring Usernames and Passwords on the CPE

Network Environment

Figure 0-6



Configuration Method

- Enable CWMP.
- On the CPE, configure the ACS username and password to be authenticated for the CPE to connect to the ACS.
- On the CPE, configure the CPE username and password to be authenticated for the ACS to connect to the CPE.

CPE

```
Ruijie# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# cwmp
Ruijie(config-cwmp)# acs username USERB
Ruijie(config-cwmp)# acs password PASSWORDB
Ruijie(config-cwmp)# cpe username USERB
Ruijie(config-cwmp)# cpe password PASSWORDB
```

Verification

- Run the **show** command on the CPE to check whether the configuration commands have been successfully applied.

CPE

```
Ruijie # show cwmp configuration
CWMP Status                     : enable
ACS URL                         : http://10.10.10.1:7547/acs
ACS username                    : USERA
ACS password                    : ******
CPE URL                         : http://10.10.10.2:7547/
CPE username                    : USERB
CPE password                    : ******
```

Configuring the URLs of the ACS and the CPE

Network Environment

See 错误!未找到引用源。.

Configuration Method

- Configure the ACS URL.
- Configure the CPE URL.

CPE

```
Ruijie# configure terminal
Ruijie(config)# cwmp
Ruijie(config-cwmp)# acs url http://10.10.10.1:7547/acs
Ruijie(config-cwmp)# cpe url http://10.10.10.1:7547/
```

Verification

Run the **show** command on the CPE to check whether the configuration commands have been successfully applied.

CPE

```
Ruijie #show cwmp configuration
CWMP Status                      : enable
ACS URL                          : http://10.10.10.1:7547/acs
ACS username                     : USERA
ACS password                     : ******
CPE URL                          : http://10.10.10.2:7547/
```

Common Errors

- The user-input encrypted password is longer than 254 characters, or the length of the password is not an even number.
- The user-input plaintext password is longer than 100 characters.
- The user-input plaintext password contains illegal characters.
- The user-input encrypted password contains illegal characters (the legitimate characters includes only 0~9, a~f and A~F)
- The URL of the ACS is set to **NULL**.
- The URL of the CPE is set to **NULL**.

Configuring CWMP-Related Attributes

Configuration Effect

- You can configure common functions of the CPE, such as the backup and restoration of its firmware or configuration file, whether to enable the CPE to download firmware and configuration files from the ACS, and whether to enable the CPE to upload its configuration and log files to the ACS.

Configuration Method

Configuring the Periodic Notification Function of the CPE

- (Optional) The value range is from 30 to 3,600 in seconds. The default value is 600 seconds.
- Perform this configuration to reset the periodical notification interval of the CPE.

Command	**cpe inform [interval** *seconds*] **[starttime** *time*]
Parameter Description	*seconds*: Specifies the periodical notification interval of the CPE. The value range is from 30 to 3,600 in seconds. *time*: Specifies the date and time for starting periodical notification in *yyyy-mm-ddThh:mm:ss* format.
Command Mode	CWMP configuration mode
Defaults	The default value is 600 seconds.
Usage Guide	Use this command to configure the periodic notification function of the CPE.

	● If the time for starting periodical notification is not specified, periodical notification starts after the periodical notification function is enabled. The notification is performed once within every notification interval. ● If the time for starting periodical notification is specified, periodical notification starts at the specified start time. For instance, if the periodical notification interval is set to 60 seconds and the start time is 12:00 am next day, periodical notification will start at 12:00 am next day and once every 60 seconds.

↘ Disabling the Function of Downloading Firmware and Configuration Files from the ACS

- (Optional) The CPE can download firmware and configuration files from the ACS by default.
- Perform this configuration if the CPE does not need to download firmware and configuration files from the ACS.

Command	**disable download**
Parameter Description	N/A
Defaults	The CPE can download firmware and configuration files from the ACS by default.
Command Mode	CWMP configuration mode
Usage Guide	Use this command to disable the function of downloading main program and configuration files from the ACS. ● This command does not act on configuration script files. The configuration scripts can still be executed even if this function is disabled.

↘ Disabling the Function of Uploading Configuration and Log Files to the ACS

- (Optional.) The CPE can upload configuration and log files to the ACS by default.
- Perform this configuration if the CPE does not need to upload configuration and log files to the ACS.

Command	**disable upload**
Parameter Description	N/A
Defaults	The CPE can upload configuration and log files to the ACS by default.
Command Mode	CWMP configuration mode
Usage Guide	Use this command to disable the function of uploading configuration and log files to the ACS.

↘ Configuring the Backup and Restoration of the Firmware and Configuration Files of the CPE

- (Optional) The backup and restoration of the firmware and configuration files of the CPE is enabled by default. The value range is from 10 to 600 in seconds. The default value is 30 seconds.
- The longer the delay-time is, the longer the reboot will be complete.
- Perform this configuration to modify the function of backing up and restoring the firmware and configuration files of the CPE.

Command	**cpe back-up [delay-time** *seconds*]
Parameter Description	*seconds*: Specifies the delay for backup and restoration of the firmware and configuration file of the CPE.
Defaults	The default value is 30 seconds.
Command	CWMP configuration mode

Mode	
Usage Guide	N/A

➘ Configuring the ACS Response Timeout

- (Optional) The value range is from 10 to 600 in seconds. The default value is 30 seconds.
- Perform this configuration to modify the ACS response timeout period on the CPE.

Command	**timer cpe- timeout** *seconds*
Parameter Description	*seconds*: Specifies the timeout period in seconds. The value range is from 10 to 600.
Defaults	The default value is 30 seconds.
Command Mode	CWMP configuration mode
Usage Guide	N/A

Verification

- Run the **show cwmp configuration** command.

Command	**show cwmp configuration**
Parameter Description	N/A
Command Mode	Privileged EXEC mode
Usage Guide	N/A
Configuration Examples	The following example displays the CWMP configuration.

```
Ruijie(config-cwmp)#show cwmp configuration
CWMP Status                   : enable
ACS URL                      : http://www.ruijie.com.cn/acs
ACS username                  : admin
ACS password                  : ******
CPE URL                      : http://10.10.10.2:7547/
CPE username                  : ruijie
CPE password                  : ******
CPE inform status              : disable
CPE inform interval             : 60s
CPE inform start time           : 0:0:0 0 0 0
CPE wait timeout               : 50s
CPE download status             : enable
CPE upload status              : enable
CPE back up status             : enable
CPE back up delay time          : 60s
```

Configuration Examples

➘ Configuring the Periodical Notification Interval of the CPE

Network Environment

See 错误!未找到引用源。.

Configuration Steps

- Enable the CWMP function and enter CWMP configuration mode.
- Set the periodical notification interval of the CPE to 60 seconds.

CPE

```
Ruijie#config
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)#cwmp
Ruijie(config-cwmp)#cpe inform interval 60
```

Verification

Run the **show** command on the CPE to check whether the configuration commands have been successfully applied.

CPE

```
Ruijie #show cwmp configuration
CWMP Status                       : enable
……
CPE inform interval              : 60s
```

Disabling the Function of Downloading Firmware and Configuration Files from the ACS

Network Environment

See 错误!未找到引用源。.

Steps

- Enable the CWMP function and enter CWMP configuration mode.
- Disable the function of downloading firmware and configuration files from the ACS.

CPE

```
Ruijie#config
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)#cwmp
Ruijie(config-cwmp)#disable download
```

Verification

Run the **show** command on the CPE to check whether the configuration commands have been successfully applied.

CPE

```
Ruijie #show cwmp configuration
CWMP Status                       : enable
……
CPE download status              : disable
```

Disabling the Function of Uploading Configuration and Log Files to the ACS

Network Environment

See 错误!未找到引用源。.

Configuration Steps

- Enable the CWMP function and enter CWMP configuration mode.
- Disable the CPE's function of uploading configuration and log files to the ACS.

CPE

```
Ruijie#config
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)#cwmp
Ruijie(config-cwmp)# disable upload
```

Verification

Run the **show** command on the CPE to check whether the configuration commands have been successfully applied.

CPE

```
Ruijie #show cwmp configuration
CWMP Status                    : enable
……
CPE upload status              : disable
```

Configuring the Backup and Restoration Delay

Network Environment

See 错误!未找到引用源。.

Configuration Steps

- Enable the CWMP function and enter CWMP configuration mode.
- Set the backup and restoration delay to 100 seconds.

CPE

```
Ruijie#config
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)#cwmp
Ruijie(config-cwmp)# cpe back-up Seconds 30
```

Verification

- Run the **show** command on the CPE to check whether the configuration commands have been successfully applied.

CPE

```
Ruijie #show cwmp configuration
CWMP Status                      : enable
……
CPE back up delay time           : 30s
```

Configuring the ACS Response Timeout of the CPE

Network Environment

See 错误!未找到引用源。.

Configuration Steps

- Enable the CWMP function and enter CWMP configuration mode.
- Set the response timeout of the CPE to 100 seconds.

CPE

```
Ruijie# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# cwmp
Ruijie(config-cwmp)# timer cpe-timeout 100
```

Verification

- Run the **show** command on the CPE to check whether the configuration commands have been successfully applied.

CPE

```
Ruijie#show cwmp configuration
CWMP Status                    : enable
……
CPE wait timeout               : 100s
```

Common Errors

N/A

Monitoring

Displaying

Command	Function
show cwmp configuration	Displays the CWMP configuration.
show cwmp status	Displays the CWMP running status.

Configuring USB/SD

Overview

This document describes usage of USB/SD storage devices (mainly U disk/SD). The system only recognizes the U-disk/SD card partitioned by FAT. Other file systems cannot be identified.

After inserting a U disk/SD, the system prompts that U disk/SD is found. The files in this U disk/SD card can be positioned and accessed through URL, such as **usb0:/abc/1.txt** or **sd0:/abc/1.txt**.

Caution

Version 10.4 (2) and the later versions support the access to U disk/SD by URL. For earlier software, use the mount path of the file system to position and access U disk/SD, such as using /mnt/usb0 to access the USB device on port 0, and using /mnt/sd to access the SD card. The mount path is prompted when the device is inserted, or is displayed when users run the **show usb** command.

The USB mobile disk (USB-HDD) is not supported.

Inserting the Device

Just insert a USB device into the USB slot. Messages as below are displayed if the system finds the device and loads the driver.

```
      *Jan  1 00:03:21: %USB-5-USB_DISK_FOUND: USB Disk<USB DISK Pro>Found!

*Jan  1 00:03:21: %USB-5-USB_DISK_PARTITION: USB_DISK_PARTITION found, /dev/uba/disc0/part4
to /mnt/usb0, size 4007657472B(3822MB)
```

Just insert an SD card into an SD slot. Messages as below are displayed if the system finds the device and loads driver.

```
*Jan  1 00:09:42: %USB-5-USB_DISK_PARTITION_MOUNT: Mount sd (type:FAT32),size :
1050673152B(1002MB)
```

SD: is the first SD partition and **size** is the partition size. This SD card has 1002 MB space.

Using the Device

After loading U disk/SD card to the system, directly run file system commands (dir, copy, del, and others) to operate U disk/SD card. Operations below show how to copy the file of U disk/SD card to flash.

Access the U disk partition.

```
Ruijie# cd usb0:/
Access the SD card partitiom.
Ruijie# cd sd0:/
```

Copy the **a.txt** file in the U disk to root directory of the device.

```
Ruijie# copy usb0:/a.txt  flash:/b.txt
```

Copy the a.txt file in the SD card to device’s the root directory of the device.

```
Ruijie# copy sd0:/a.txt  flash:/b.txt
```

Run the **dir** command. The result shows that the b.txt file has been added to the USB/SD card.

For other operation commands, see the “File System Management” section.

Caution

If there are multiple partitions in U disk/SD card, only the first FAT partition can be accessed through the device.

Note

Only the version 10.4(2) and the later versions allow users to access U disc/SD card by URL. For the earlier versions, use path to position and access the device. For example:
Access the U disk partition:
Ruijie# cd /mnt/usb0
Access the SD card partition:
Ruijie# cd /mnt/sd0

Copy the a.txt file under root directory to U disk.
Ruijie# **copy** flash:/a.txt usb0:/a.txt
Copy the a.txt file under root directory to SD card.
Ruijie# **copy** flash:/a.txt sd0:/a.txt

Showing USB Device/SD Card Information

Command	Function
Ruijie# **show usb**	Shows the USB device information of the system
Ruijie# **show sd**	Shows the SD device information of the system

In CLI command mode, use the **show usb** or the **show sd** command to view the USB / SD device information of the system. The displayed information is as follows:

```
Ruijie# show usb
Device: Mass Storage:
ID: 0
URL prefix: usb0
Disk Partitions:
usb0(type:FAT32)
```

```
Size : 131,072,000B(125MB)
Available size: 1,260,020B (1.2MB)

Ruijie# show sd

Device: Mass Storage:
ID: 1
URL prefix: sd0
Disk Partitions:
SD(type:FAT32)

Size : 131,072,000B(125MB)
Available size: 1,260,020B (1.2MB)
```

USB Mass Storage Device is the name of the device.

URL means which prefix can be used by U disk/SD card to access U disk/SD card.

Size means the available space in U disk/SD card that can be accessed.

Available size means the remaining space in U disk/SD card.

Unplugging USB Device/SD Card

Before pulling out USB device/SD card, run the command on the CLI to uninstall the device in case system is using the USB device/SD card to avoid an error.

Command	**Function**
Ruijie# **usb remove** *Device_ID*	Uninstalls the USB device with ID Device_ID

Command	Function
Ruijie# **sd remove** *Device_ID*	Uninstalls the SD device with ID Device_ID

As shown above, IDO indicates a USB device, and ID1 indicates an SD card. The commands below can uninstall the corresponding USB device and SD card.

```
Ruijie# usb remove 0
After the uninstall command is used, the system will print:
OK, now you can pull out the device 0.
*Jan  1 00:18:16: %USB-5-USB_DISK_REMOVED: USB Disk <Mass Storage> has been removed from USB
port 0!
Ruijie# sd remove 1
After the uninstall command is used, the system will print:
OK, now you can pull out the device 1
Now, you can pull out the USB device/SD card.
```

Sometimes, it may lead to failure to uninstall the device for the device is being used. Wait a while, and then run the uninstall command to pull out the device.

Caution Be sure to uninstall the device first and then unplug the device to prevent the system error.

USB/SD Faults

When the system prints the following message:

```
*Jan 2 00:00:39: %USB-3-OHCI_ERR: USB1.0 controller is not available now.
```

USB/SD 1.0 controller is not available, while USB/SD card 2.0 is still available. In this case, reset the whole system to use corresponding version U disk/SD card.

When the system prints the following message:

```
*Jan 2 00:00:39: %USB-3-EHCI_ERR: USB2.0 controller is not available now.
```

USB/SD 2.0 controller is not available, while U disc/SD card 1.0 is still available. In this case, reset the whole system to use corresponding version U disk/SD card.

Configuring System Management

Basic System Management

Showing CPU Utilization

Use the **show cpu** command to show the total CPU utilization of the system and the CPU utilization of each process:

Command	Function
Ruijie# **show cpu**	Shows CPU utilization.

By default, the device name is **Ruijie**.

The following example shows the output result of this command.

```
Ruijie#show cpu
=======================================
    CPU Using Rate Information
CPU utilization in five seconds: 25%
CPU utilization in one minute: 20%
CPU utilization in five minutes: 10%
  NO  5Sec   1Min  5Min   Process
  0    0%    0%    0%    LISR INT
  1    7%    2%    1%    HISR INT
  2    0%    0%    0%    ktimer
  3    0%    0%    0%    atimer
  4    0%    0%    0%    printk_task
  5    0%    0%    0%    waitqueue_process
  6    0%    0%    0%    tasklet_task
  7    0%    0%    0%    kevents
  8    0%    0%    0%    snmpd
  9    0%    0%    0%    snmp_trapd
 10    0%    0%    0%    mtdblock
 11    0%    0%    0%    gc_task
 12    0%    0%    0%    Context
 13    0%    0%    0%    kswapd
 14    0%    0%    0%    bdflush
 15    0%    0%    0%    kupdate
 16    0%    3%    1%    ll_mt
 17    0%    0%    0%    ll main process
 18    0%    0%    0%    bridge_relay
 19    0%    0%    0%    d1x_task
 20    0%    0%    0%    secu_policy_task
 21    0%    0%    0%    dhcpa_task
```

```
22    0%    0%    0%   dhcpsnp_task
23    0%    0%    0%   igmp_snp
24    0%    0%    0%   mstp_event
25    0%    0%    0%   GVRP_EVENT
26    0%    0%    0%   rldp_task
27    0%    2%    1%   rerp_task
28    0%    0%    0%   reup_event_handler
29    0%    0%    0%   tpp_task
30    0%    0%    0%   ip6timer
31    0%    0%    0%   rtadvd
32    0%    0%    0%   tnet6
33    2%    0%    0%   tnet
34    0%    0%    0%   Tarptime
35    0%    0%    0%   gra_arp
36    0%    0%    0%   Ttcptimer
37    8%    1%    0%   ef_res
38    0%    0%    0%   ef_rcv_msg
39    0%    0%    0%   ef_inconsistent_daemon
40    0%    0%    0%   ip6_tunnel_rcv_pkt
41    0%    0%    0%   res6t
42    0%    0%    0%   tunrt6
43    0%    0%    0%   ef6_rcv_msg
44    0%    0%    0%   ef6_inconsistent_daemon
45    0%    0%    0%   imid
46    0%    0%    0%   nsmd
47    0%    0%    0%   ripd
48    0%    0%    0%   ripngd
49    0%    0%    0%   ospfd
50    0%    0%    0%   ospf6d
51    0%    0%    0%   bgpd
52    0%    0%    0%   pimd
53    0%    0%    0%   pim6d
54    0%    0%    0%   pdmd
55    0%    0%    0%   dvmrpd
56    0%    0%    0%   vty_connect
57    0%    0%    0%   aaa_task
58    0%    0%    0%   Tlogtrap
59    0%    0%    0%   dhcp6c
60    0%    0%    0%   sntp_recv_task
61    0%    0%    0%   ntp_task
62    0%    0%    0%   sla_deamon
63    0%    3%    1%   track_daemon
64    0%    0%    0%   pbr_guard
65    0%    0%    0%   vrrpd
66    0%    0%    0%   psnpd
```

```
 67    0%    0%    0%    igsnpd
 68    0%    0%    0%    coa_recv
 69    0%    0%    0%    co_oper
 70    0%    0%    0%    co_mac
 71    0%    0%    0%    radius_task
 72    0%    0%    0%    tac+_acct_task
 73    0%    0%    0%    tac+_task
 74    0%    0%    0%    dhcpd_task
 75    0%    0%    0%    dhcps_task
 76    0%    0%    0%    dhcpping_task
 77    0%    0%    0%    dhcpc_task
 78    0%    0%    0%    uart_debug_file_task
 79    0%    0%    0%    ssp_init_task
 80    0%    0%    0%    rl_listen
 81    0%    0%    0%    ikl_msg_operate_thread
 82    0%    0%    0%    bcmDPC
 83    0%    0%    0%    bcmL2X.0
 84    3%    3%    3%    bcmL2X.0
 85    0%    0%    0%    bcmCNTR.0
 86    0%    0%    0%    bcmTX
 87    0%    0%    0%    bcmXGS3AsyncTX
 88    0%    2%    1%    bcmLINK.0
 89    0%    0%    0%    bcmRX
 90    0%    0%    0%    mngpkt_rcv_thread
 91    0%    0%    0%    mngpkt_recycle_thread
 92    0%    0%    0%    stack_task
 93    0%    0%    0%    stack_disc_task
 94    0%    0%    0%    redun_sync_task
 95    0%    0%    0%    conf_dispatch_task
 96    0%    0%    0%    devprob_task
 97    0%    0%    0%    rdp_snd_thread
 98    0%    0%    0%    rdp_rcv_thread
 99    0%    0%    0%    rdp_slot_change_thread
100    4%    2%    1%    datapkt_rcv_thread
101    0%    0%    0%    keepalive_link_notify
102    0%    0%    0%    rerp_msg_recv_thread
103    0%    0%    0%    ip_scan_guard_task
104    0%    0%    0%    ssp_ipmc_hit_task
105    0%    0%    0%    ssp_ipmc_trap_task
106    0%    0%    0%    hw_err_snd_task
107    0%    0%    0%    rerp_packet_send_task
108    0%    0%    0%    idle_vlan_proc_thread
109    0%    0%    0%    cmic_pause_detect
110    1%    1%    1%    stat_get_and_send
111    0%    1%    0%    rl_con
```

```
112   75%   80%   90%   idle
```

As shown in the above list, the first three lines indicate the total CPU utilization in the last 5 seconds, 1 minute, and 5 minutes respectively, including the CPU utilization of LISRs, HISRs and tasks, followed by the CPU utilization of various processes. The parameters in the columns are described as follows:

- **No**: number
- **5Sec**: CPU utilization in the last 5 seconds
- **1Min**: CPU utilization in the last 1 minute
- **5Min**: CPU utilization in the last 5 minutes
- **Process**: process name

The first two lines indicate the CPU utilization of all LISRs and all HISRs respectively. All the lines starting from the third line indicate the CPU utilization of processes. The last line indicates the CPU utilization of idle processes. As with **System Idle Process** in the Windows operating system, it indicates an idle status. The above example shows that the CPU utilization of idle processes in the last 5 seconds is 75%, meaning that 75% of the CPU resources are available.

Configuring CPU Utilization Log Thresholds

Use the following command to configure CPU utilization log thresholds.

Command	Function
cpu-log log-limit *low_num high_num*	Configures the high and low thresholds for triggering CPU utilization logs.

By default, the high threshold is 100% and the low threshold is 90%.

The following example sets the low threshold to 70% and the high threshold to 80%.

```
Ruijie# configure terminal                  // Enter the global configuration mode.
Ruijie(config)# cpu-log log-limit 70 80    // Configure the thresholds for triggering CPU logs.
```

If the CPU utilization is higher than 80%, the following information is displayed:

Oct 20 15:47:01 %SYSCHECK-5-CPU_USING_RATE_LOG: CPU utilization rate in one minute: 95%. rl_con occupied most CPU utilization rate: 94%.

If the CPU utilization is lower than 70%, the following information is displayed:

Oct 20 15:47:01 %SYSCHECK-5-CPU_USING_RATE_LOG: CPU utilization rate in one minute: 68%. rl_con occupied most CPU utilization rate: 60%.

Oct 20 15:47:01 %SYSCHECK-5-CPU_USING_RATE_LOG: The CPU utilization ratio has been decreased.

Configuring System Memory Display

Showing the Usage of System Memory

Use the **show memory** command to show the usage and status of system memory:

Command	Function
Ruijie# **show memory**	Shows the usage of system memory.

The switch name is Ruijie by default.

Below is the result of executing this command:

```
Ruijie#show memory
System Memory Statistic:
  Free pages: 174164
    watermarks : min 2012, lower 4024, low 6036, high 9048
  System Total Memory : 1024MB, Current Free Memory : 740580KB
  Used Rate : 29%
```

The above information includes:

Parameter		Description
Free pages		The total free pages of all areas.
watermarks	min	Memory resources are extremely insufficient. It can only keep the kernel running. All application modules fails to run if the minimum watermark has been reached.
	lower	Memory resources are severely insufficient. One route protocol will auto-exit and release the memory if the lower watermark has been reached. For the details, see the **memory-lack exit-policy** command.
	low	Memory resources are insufficient. The route protocol will be in OVERFLOW state if the low watermark has been reached. In the overflow state, the routers do not learn new routes any more. The commands are not allowed to be executed when the memory lacks.
	high	There is plenty of memory resources. Each route protocol restores the state from OVERFLOW to normal.
System Total Memory		System total memory
System Free Memory		System free memory, including free pages space and all free space in the cache pool
Used Rate		Memory utilization rate

Configuring MIB

MIB Lists

The followings are the supported standard MIBs:

- BRIDGE-MIB (RFC1493)
- EtherLike-MIB(RFC1643)
- IF-MIB(RFC2863)
- RFC1213-MIB
- RMON1-MIB(supports RMON etherStats, etherHistory,alarms, events)
- SNMPv2-MIB
- SNMPv3-MIB(supports USM, VACM)

The followings are the private MIBs:

- RUIJIE-AAA-MIB
- RUIJIE -ENTITY-MIB
- RUIJIE -RIP-MIB
- RUIJIE -MEMORY-MIB

You can use the **show snmp mib** command to view the supported MIBs in the current system:

```
Ruijie# show snmp mib
sysDescr
sysObjectID
sysUpTime
sysContact
sysName
sysLocation
sysServices
sysORLastChange
ifNumber
ifEntry
ifEntry.ifIndex
ifEntry.ifDescr
ifEntry.ifType
ifEntry.ifMtu
ifEntry.ifSpeed
ifEntry.ifPhysAddress
ifEntry.ifAdminStatus
ifEntry.ifOperStatus
ifEntry.ifLastChange
ifEntry.ifInOctets
ifEntry.ifInUcastPkts
ifEntry.ifInNUcastPkts
ifEntry.ifInDiscards
ifEntry.ifInErrors
ifEntry.ifInUnknownProtos

ifEntry.ifOutOctets

ifEntry.ifOutUcastPkts

ifEntry.ifOutNUcastPkts

ifEntry.ifOutDiscards

ifEntry.ifOutErrors

ifEntry.ifOutQLen

ifEntry.ifSpecific

......
```

Obtaining the MIB Files

You can obtain the standard MIB information on the http://ietf.org/rfc.html.
You can obtain the private MIB information on the website.

Configuring One-click Upgrade

Understanding One-click Upgrade

Basic Concepts

One-click upgrade means that a user presses the **FUNC** key on the device panel to reset the system. Before the system reset, the user can individually upgrade the software version or configuration file by using the files stored in an SD card or USB drive or upgrade the software version and configuration file at the same time. Specifically, one-click upgrade provides the following functions:

- One-click system reset
- One-click upgrade of software version and configuration file

One-click System Reset

When no SD card or USB drive is inserted into the device or when an SD card or a USB drive is inserted but no appropriate software version or configuration file is available in the SD card or USB drive for upgrade, the user resets the system by pressing the **FUNC** key to the original software version.

One-click Upgrade of Software Version and Configuration File

When an SD card or a USB drive is inserted into the device and the user presses the **FUNC** key, the system scans for the installation package and configuration file in turn in the root directory of the SD card or USB drive. After finding the installation package and configuration file, the system first performs a validity check and then upgrades the device by using the installation package and configuration file if the installation package and configuration file are valid. After the upgrade is complete, the system resets by using the new software version and configuration file.

Though old line cards do not provide the **FUNC** key, so long as the corresponding software is available, one-click upgrade of the software version and configuration file can be triggered through a configuration command. The upgrade triggered by using a configuration command is the same as that by pressing the **FUNC** key. The system scans for the installation package and configuration file in turn in the root directory of the SD card or USB drive. After finding the installation package and configuration file, the system first performs a validity check and then upgrades the device by using the installation package and configuration file if the installation package and configuration file are valid. After the upgrade is successful, the system resets.

Note

When scanning the inserted storage media, the system scans the SD card in preference. If no SD card is inserted or the SD card does not store an appropriate installation package or configuration file, the system scans the USB drive. If the inserted SD card and USB drive do not store an appropriate installation package or configuration file, the software version and configuration file of the device are not upgraded and only the system resets. The system still starts by using the original software version and configuration file.

Caution

- The installation package supports the following four filename formats:
  - *rgos.bin*
  - *Product name_project name_serial number_install.bin*

    Caution For example: RSR77_10.4(3b21)_R166400_install.bin
  - *P product name V1_project name_serial number_install.bin*

    Caution For example: PRSR77V1_10.4(3b21)_R166400_install.bin
  - *P product name v1-version number-install.bin*

    Caution For example: prsr77v1-101939-install.bin
- A product name consists of digits, uppercase, lowercase, and '-'; a version number consists of digits; a project name consist of digits, lowercase, '(', and ')'; a serial number consists of "R" and the version number. The product name in an old installation package is case-insensitive; in a new installation package, the product name is case-sensitive and the configuration file is provided in the form of an encrypted file, for example, **config.des**.
- If a **rgos.bin** file exists, the system uses the file in preference; otherwise, the system uses the first scanned installation package that meets the naming rules in preference.
- The system scans only for the installation package and configuration file in the root directory of the SD card or USB drive and always scans the SD card in preference.
- It is recommended that only one installation package be stored in the root directory of the SD card of USB drive. If multiple installation packages are stored, the system randomly scans for the installation package and takes the first scanned installation package for upgrade, which cannot ensure the correct sequence of searching for the software version.
- The installation package and configuration file must be stored on the same storage medium. If the installation package and configuration file are located in the SD card and USB drive respectively, only the file in the SD card is upgraded.
- Before upgrading the configuration file, you must ensure that the flash memory has sufficient space for storing the configuration file to be upgraded. The size of the configuration file depends on the configuration and is generally not smaller than 1 MB.
- Before upgrading the installation package, you must ensure that the flash memory has sufficient space for storing the installation package to be upgraded. The size of the installation package is generally not smaller than 30 MB.
- Before performing the **FUNC** key triggered one-click upgrade, you must ensure that all line cards are started, that is, the **SYS** indicators of boards become green. For specific positions of **SYS** indicators, see the *Hardware Installation Manual*.
- When using one-click upgrade based on the **FUNC** key, you must ensure that an SD card or USB drive is inserted and the device detects the SD card or USB drive, that is, the indicator of the SD card or USB drive becomes green. For the specific position of the indicator, see the *Hardware Installation Manual*.

- After one-click upgrade is complete, the device resets automatically. If one-click upgrade fails, the indicator of the SD card or USB drive is off and the status indicator becomes red for three seconds. No matter whether the device is upgraded successfully or not, if the SD card or USB drive stores a file that can be used for upgrade, the system generates a Syslog file whose format is “product name-update.log” in the root directory of the SD card or USB drive.
- The function of triggering one-click upgrade through a configuration command requires the software version to be 10.4(3b21) or later.

When scanning the inserted storage media, the system scans the SD card in preference. If no SD card is inserted or the inserted SD card does not store an appropriate installation package or configuration file, the system scans the USB drive. If the inserted SD card and USB drive do not have an appropriate installation package or complete configuration file, the software version and configuration file of the device are not upgraded and only the system resets. The system still starts by using the original software version and configuration file.

Encryption of Configuration File

The configuration file contains confidential information of the user. Therefore, security protection measures must be taken to avoid disclosure of the information. Ruijie’s one-click upgrade function provides a tool on the PC side to realize file encryption. That is, the configuration file is encrypted during transmission. When the device performs one-click upgrade, the device reads the encrypted configuration file in the SD card or USB drive and decrypts the configuration file. After successfully decrypting the configuration file, the device upgrades the configuration file; when the device fails to decrypt the configuration file, the configuration file is used incorrectly or the file is damaged during transmission. In this case, the device does not upgrade the configuration file.

For instructions on the encryption tool on the PC side for Ruijie’s one-click upgrade function, see the *PC Client Configuration for One-click Upgrade*.

Backup of Configuration File

Upgrading the configuration file is an irreversible operation. If you find an error in upgrading the configuration file and want to restore to the previous configuration, you must upgrade the configuration file again. To help the user obtain the configuration file before upgrade, before one-click upgrade is performed, the configuration file is encrypted and stored in an SD card or USB drive as follows: root directory/update_backup/date_time_config.des (if the same file is found in the directory, the file is overwritten). The configuration file of the device is backed up only when the SD card or USB drive stores the available installation package or configuration file. If the SD card or USB drive does not store any installation package or configuration file for upgrade, the configuration file of the device is not backed up. Before using this function, ensure that the SD card has enough space to store the configuration file. The required storage space depends on the configuration and is generally not smaller than 1 MB.

When you find that the configuration file is upgraded incorrectly, copy the **date_time_config.des** file in **/update_backup** under the root directory of the device to the root directory of the SD card or USB drive, rename it **config.des**, and perform one-click upgrade once again to restore the previous configuration.

The configuration file that is backed up can be used by only the device that generates the configuration file to ensure that the backed up configuration file is not incorrectly used on other devices.

Synchronization of Configuration File

The configuration file running on the device must be the same as the configuration file stored in the SD card or USB drive so that the device is replaced directly and the network is restored quickly for the user by using one-click upgrade when the device is faulty. To meet this requirement, when you modify and save the configuration, the device automatically encrypts the configuration file and synchronizes it to the **config.des** file in the root directory of the SD card. If the file already exists in the root directory, the file is overwritten. Before applying this function, ensure that the SD card has enough space to store the configuration file. The required storage space depends on the configuration and is generally not smaller than 1 MB.

The configuration file is synchronized only when you save the configuration. When you modify the device configuration in other modes (for example, rename the existing configuration file and copy other configuration files to the flash memory), the device does not synchronize the configuration file.

The synchronized configuration file can be used on any device that supports one-click upgrade, but you need to ensure that the configuration is applicable to the device.

Synchronization of Installation Package

The configuration file running on the device must be consistent with the configuration file stored in the SD card or USB drive so that the device can be replaced directly and the network can be restored quickly by using one-click upgrade when the device is faulty. To meet this requirement, when you upgrade the installation package, the device automatically synchronizes the installation package to the **rgos.bin** file in the root directory of the SD card. If the file already exists in the root directory, the file is overwritten. Before using this function, ensure that the SD card has enough space to store the installation package. The required storage space depends on the size of the installation package and is generally not smaller than 50 MB.

The installation package is synchronized only when you run the **copy tftp** command. When you upgrade the installation package in other modes (for example, rename the existing installation package and copy files in other SD cards or USB drives to the flash memory), the device does not synchronize the installation package.

Working Principle

The **FUNC** key is available on the device panel. If you press the **FUNC** key, the system interrupts the current task and executes the corresponding task. During execution of the task triggered by the **FUNC** key, the system checks whether an SD card or a USB drive is inserted into the current device. If not, the system resets directly. Otherwise, the system first checks whether an SD card is inserted prior to a USB drive. Each time when the system scans a storage medium (SD card or USB drive), the system checks whether an installation package matching the specified file format exists in the root directory of the storage medium and whether the **config.des** file exists in the root directory. If the file exists, the system decrypts the file and performs a validity check on the file. If the check on the installation package and configuration file is successful, the system upgrades the software version and configuration file of the device and then resets by using the new software version and configuration file after the upgrade is complete. If no applicable installation package or configuration file exists or the installation package and configuration file do not pass the validity check, the system does not upgrade the installation package or configuration file and directly resets by using the original software version and configuration file.

According to the preceding description, in addition to the default one-click system reset based on the **FUNC** key, the one-click upgrade function also supports upgrading the software version and configuration file of a device by using a new software version and configuration file stored in an SD card or a USB drive. This provides another convenient means for the user to reset the system and upgrade the software version.

Different from the traditional mode in which the user logs in to the device through the console port or telnet to perform reset and upgrade, one-key upgrade enables users to reset the device and upgrade the software version and configuration file, without knowing fundamental knowledge about the device. Therefore, users in remote areas without any technical basis can operate and use access routers. When the original software version and configuration file need to be upgraded, only an SD card or a USB drive storing the installation package and configuration file of the new software version needs to be delivered. The customer even without knowing fundamental knowledge about the device can perform the operation and technical personnel do not need to operate on site. This greatly saves the cost of maintenance and the time of technical personnel.

Protocols and Standards

None

Default Configuration

None

Configuring One-click Upgrade

Triggering One-click Upgrade by Using a Command

Command	Function
Ruijie#**onekey**	Enables one-click upgrade.

☑ RSR77 series of version 10.4(3b21) and later support this function.

Displaying Configuration

None

Configuration Example 1

One-click system reset

Networking Requirements

None

Networking Topology

None

Configuration Tips

None

Configuration Steps

Press the **FUNC** key to reset the system when all the **SYS** indicators of boards are green.

Verification

After the system resets, run the **show version** command to check whether the software version of the system is the same as that before the system resets.

Configuration Example 2

One-click upgrade of software version and configuration file

Networking Requirements

None

Networking Topology

None

Configuration Tips

None

Configuration Steps

1) Insert the SD card or USB drive storing the installation package of the new software version into the device (pay attention to the file format. For details, see "One-click Upgrade of Software Version and Configuration File").

Press the **FUNC** key when the **SYS** indicators of all boards are green. The system performs software upgrade and then resets.

Verification

If the SD card or USB drive stores an appropriate configuration file or installation package and the upgrade is successful, the device resets automatically after the upgrade is complete.

If the SD card or USB drive does not store an appropriate configuration file or installation package, as a result of which the upgrade fails, the **SYS** indicators of the device become red for three seconds and then the device resets. For specific positions of **SYS** indicators, see the *Hardware Installation and Reference Guide*.

After the system resets, run the **show version** command to check whether the software version of the system is updated.

Configuring Debugging Improvement

Overview

Debugging information of devices, including output information of syslog, show commands and debug commands, can help post-sales or R&D engineers fast locate problems. However, it is inconvenient to use debugging information at present. For example, syslog information is buffered only in the memory. Once upon exception restart, previously saved syslog information will be lost, which is unfavorable to troubleshooting. In addition, a command for collecting all fault information is unavailable. This document describes how to configure debugging improvement based on the preceding problems.

One-key Collection

One-key collection comprises two parts:

The CLI **show tech-support** command can be used to display platform information in all scenarios.

The **show tech-support** <*module name*> command can be used to display module information in certain scenarios.

One-key Collection of Platform Information

Run the following command for one-key collection of platform information:

Command	Description
Ruijie# **show tech-support**	Collects platform information using the one-key mode.

At present, the show command for one-key collection of platform information integrates the following commands:

1. System platform

```
------------------ show version ------------------

------------------ show running-config ------------------

------------------ show user ------------------

------------------ dir ------------------

------------------ show memory------------------

------------------ show cpu ------------------

------------------ show task statistic------------------

------------------ show timer statistic------------------

------------------ show queue statistic-----------------

------------------ show skbuffer statistic--------------

------------------ show environment----------------------
```

------------------ show version slot----------------------

------------------ show clock----------------------

------------------ show exception----------------------

2. Express forwarding platform

------------------ show core ------------------

------------------ show efb ----------------------

------------------ show core interface all----------------------

3. Flow platform

------------------ show ip fpm statistics ----------------------

------------------ show ip fpm counters----------------------

------------------ show ipv6 fpm statistics----------------------

4. Routing module

------------------ show ip route summary all----------------------

------------------ Show ip route count----------------------

------------------ show ip ref route statistic----------------------

------------------ show ip ref adjacency statistic------------------

------------------ show ipv6 ref route statistic ----------------------

------------------ show ipv6 route summary all------------------

------------------ show ipv6 ref adjacency statistic------------------

5. Interface

------------------ show interfaces ------------------

------------------ show ip interface brief ------------------

------------------ show ef-interfaces ------------------

------------------ show controllers e1 ------------------

In addition, the show command for platform information integrates the show commands for all modules. For the collection of module information, see the next section.

One-key Collection of Module Information

Run the following command for one-key collection of module information:

Command	Description
Ruijie# **show tech-support** <*module name*>	Collects information of modules using the one-key mode.

At present, the command can be used to collect information of the following modules:

3G/CE1/MLP/PPP/NAT/IPSEC/L2TP/BFD/QOS/RIP/OSPF/BGP/ETHERNET/MSTP/FR/PPPOE/CA/DLDP/MPLS

The show commands to be integrated for the modules are as follows:

3G:

show cell info

show plmn backup

show dialer

CE1:

show interface

show controller e1

MLP:

show ppp mul

show ref ppp mul

NAT:

show ip nat translations

show ip nat statistics rule nouse/syn

IPSec:

show crypto isakmp policy

show crypto ipsec sa

show crypto ipsec transform-set

show crypto map

show crypto acl

show crypto detail

show crypto state

L2TP:

show vpdn

show vpdn tunnel

show vpdn session

BFD:

show bfd neighbor details

QoS:

show rate-limit

show policy-map

show queue cq/pq/rtpq/wfq

show traffic-shape queue

show traffic-shape statistics

RIP:

show ip protocol

show ip route

show ip rip

show ip rip database

show ip rip interface

show ip rip peer

show ip rip external

show ip route summary

OSPF:

show ip protocol

show ip route

show ip ospf neighbor detail

show ip ospf interface

show ip ospf database

show ip ospf sham-links

show ip ospf virtual-links

show ip ospf spf

show ip ospf topo

show ip ospf border-routers

show ip ospf summary-address

show ip ospf route

show ip ospf route summary

BGP:

show ip protocol

show ip route

show ip bgp neighbor

show ip bgp summary

show ip bpg all summary

show bgp all summary

Ethernet/MSTP:

show arp

FR:

show fr map

show fr lmi

show fr pvc

PPPoE:

show pppoe session

CA:

show crypto pki trustpoints

show crypto pki certificates

DLDP:

show dldp interface

MPLS:

show ip vrf interfaces

show ip ref adjacency

show ip ref mpls packet debug-buf

show mpls forwarding-table

Configuring Flow Platform

Understanding the Flow Platform

Overview

The flow platform achieves a perfect combination of services and performance, because services (such as QoS, ACL, NAT, and PBR) enabled on a service processing board have nearly no impact on forwarding performance.

Basic Concepts

Service packets based on Layer 3 usually can be abstracted as flows. A flow identifies a sequence of packets from a specific source to a specific destination. Generally, a flow is indentified by a sextuplet, which includes a source address, a destination address, a source port ID, a destination port ID, a transport layer protocol, and VRF.

Working Principle

When the first packet is forwarded through an entire routing process and service processing, the sextuplet of the packet identifies a flow. If a packet received later matches the sextuplet, it is forwarded in the same way as the first packet. The sextuplet and an outbound interface of the packet compose a flow entry. Packets that match the flow entry are forwarded directly without experiencing the entire routing process or service processing any longer. Therefore, enabled services (such as QoS, ACL, NAT, and PBR) have nearly no impact on forwarding performance.

Protocols and Standards

None.

Typical Application

When services (such as QoS, ACL, NAT, and PBR) are enabled, the flow platform is automatically enabled to accelerate service processing.

Configuring the IPv4 Function of the Flow Platform

Default Configuration

The following table describes the default configuration of the flow platform.

Feature	Default Setting
Flow platform	Disabled

Flow overflow alarm interval of the flow platform	5 seconds
Flow overflow alarm threshold of the flow platform	95%
Maximum number of flow entries in the IPv4 flow table	180,223

Prerequisites

When service modules (such as QoS, ACL, NAT, and PBR service modules) that rely on the flow platform are configured, the flow platform is automatically enabled.

Configuration Steps

Step	Configuration Task	Description
Step 1	Configure service modules.	(Mandatory) You can configure service modules such as QoS, ACL, NAT, and PBR.
Step 2	Configure the maximum number of flow entries in the IPv4 flow table.	(Optional) This step is performed if you need to change the memory occupied by the flow table.
Step 3	Configure the IPv4 flow overflow alarm interval of the flow platform.	(Optional) This step is performed if you need to change the flow overflow alarm interval.
Step 4	Configure the IPv4 flow overflow alarm threshold of the flow platform	(Optional) This step is performed if you need to change the flow overflow alarm threshold.

Configuring ACL to Enable the Flow Platform

Command	Function

Ruijie(config)# **access-list** *id* { **deny | permit** } { src *src-wildcard* | **host** *src* | **any** } [**time-range** *tm-rng-nam e*] [**log**]	Defines an ACL.
Ruijie(config)# **interface** *interface*	Specifies the interface to which the ACL is applied.
Ruijie(config-if)# **ip access-group** *id* { in | out } [**unreflect**]	Applies the ACL to the specific interface.

The following example enables the ACL function on the port GigabitEthernet 0/1.

```
Ruijie# configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

Ruijie(config)# access-list 101 permit tcp 192.168.12.0 0.0.0.255 any

Ruijie(config)# interface GigabitEthernet 0/1

Ruijie(config-if)# ip address 192.168.12.1 255.255.255.0

Ruijie(config-if)# ip access-group 101 in
```

Configuring the Maximum Number of Flow Entries in the IPv4 Flow Table

Command	**Function**
Ruijie(config)# **ip fpm flow max-entries** *flow-number*	Configures the maximum number of flow entries in the IPv4 flow table.

The following example configures the maximum number of IPv4 flow entries as 120,000.

```
Ruijie# configure terminal

Enter configuration commands, one per line.  End with CNTL/Z.

Ruijie(config)# ip fpm flow max-entries 120000

FPM subsystem is reinitializing...

Ruijie(config)#*Oct  6 17:35:21: %FPM-5-RESTARTED: The device IPv4 flow max-entries changed.
```

Configuring the IPv4 Flow Overflow Alarm Interval of the Flow Platform

Command	Function
Ruijie(config)# **ip fpm flow alert interval** *seconds*	Configures the IPv4 flow overflow alarm interval of the flow platform.

The following example configures the IPv4 flow overflow alarm interval of the flow platform as 120s.

```
Ruijie# configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

Ruijie(config)# ip fpm flow alert interval 120
```

Configuring the IPv4 Flow Overflow Alarm Threshold of the Flow Platform

Command	Function
Ruijie(config)# **ip fpm flow alert threshold** *percent-value*	Configures the IPv4 flow overflow alarm threshold of the flow platform.

The following example configures the IPv4 flow overflow alarm threshold of the flow platform as 80%.

```
Ruijie# configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

Ruijie(config)# ip fpm flow alert threshold 80
```

Configuring the IPv6 Function of the Flow Platform

Default Configuration

The following table describes the default configuration of the flow platform.

Feature	Default Setting
Flow platform	Disabled

Flow overflow alarm interval of the flow platform	5 seconds
Flow overflow alarm threshold of the flow platform	95%
Maximum number of flow entries in the IPv6 flow table	81,920

Prerequisites

When service modules (such as QoS v6, ACLv6, NAT64, and PBRv6 service modules) that rely on the flow platform are configured, the flow platform is automatically enabled.

Configuration Steps

Step	Configuration Task	Description
Step 1	Configure service modules.	(Mandatory) You can configure service modules such as QoS v6, ACLv6, NAT64, and PBRv6.
Step 2	Configure the maximum number of flow entries in the IPv6 flow table.	(Optional) This step is performed if you need to change the memory occupied by the flow table.
Step 3	Configure the IPv6 flow overflow alarm interval of the flow platform.	(Optional) This step is performed if you need to change the flow overflow alarm interval.
Step 4	Configure the IPv6 flow overflow alarm threshold of the flow platform.	(Optional) This step is performed if you need to change the flow overflow alarm threshold.

Configuring ACLv6 to Enable the Flow Platform

Command	Function
Ruijie(config)# **ipv6 access-list** *name*	Enters ACL configuration mode.

Ruijie (config-ipv6-nacl)#[*sn*] {**permit** | **deny** } *port* { *src-ipv6-prefix/prefix-len* | **host** *src-ipv6-addr* | **any** } { *dst-ipv6-pfix/pfix-len* | **any** | **host** *dst-ipv6-addr* } [**dscp** *dscp*] [**flow-label** *flow-label*] [**fragments**] [**range** lower upper] [**time-range***tm-rng-name*]	Adds an entry to the ACL. For details about the command, see the Command Reference.
Ruijie(config-exp-nacl)# **exit**	Exits ACL configuration mode.
Ruijie(config)# **interface** *interface*	Specifies the interface to which the ACL is applied.
Ruijie(config-if)# **ipv6 traffic-filter** *name* { **in** | **out** }	Applies the ACL to the specific interface.

The following example enables the ACL function on the port GigabitEthernet 0/1.

```
Ruijie# configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

Ruijie(config)# ipv6 access-list v6-list

Ruijie(config-ipv6-acl)# permit ipv6 2001:db8:1::1/64 any

Ruijie(config-ipv6-acl)# deny ipv6 any any

Ruijie(config-ipv6-acl)# exit

Ruijie(config)# interface GigabitEthernet 0/1

Ruijie(config-if)# ipv6 traffic-filter v6-list in
```

Configuring the Maximum Number of Flow Entries in the IPv6 Flow Table

Command	**Function**
Ruijie(config)# **ipv6 fpm flow max-entries** *flow-number*	Configures the maximum number of flow entries in the IPv6 flow table.

The following example configures the maximum number of flow entries in the IPv6 flow table as 70,000.

```
Ruijie# configure terminal
```

```
Enter configuration commands, one per line. End with CNTL/Z.

Ruijie(config)# ipv6 fpm flow max-entries 70000

FPM subsystem is reinitializing...

Ruijie(config)#*Oct  6 17:35:21: %FPM-5-RESTARTED: The device IPv6 flow max-entries changed.
```

Configuring the IPv6 Flow Overflow Alarm Interval of the Flow Platform

Command	**Function**
Ruijie(config)#**ipv6 fpm flow alert interval** *seconds*	Configures the IPv6 flow overflow alarm interval of the flow platform.

The following example configures the IPv6 flow overflow alarm interval of the flow platform as 120s.

```
Ruijie# configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

Ruijie(config)#ipv6 fpm flow alert interval 120
```

Configuring the IPv6 Flow Overflow Alarm Threshold of the Flow Platform

Command	**Function**
Ruijie(config)# **ipv6 fpm flow alert threshold** *percent-value*	Configures the IPv6 flow overflow alarm threshold of the flow platform.

The following example configures the IPv6 flow overflow alarm threshold of the flow platform as 80%.

```
Ruijie# configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

Ruijie(config)# ipv6 fpm flow alert threshold 80
```

Monitoring and Maintaining the Flow Platform

Command	Function
Ruijie#**clear ip fpm counters**	Clears IPv4 packet statistics of the flow platform.
Ruijie#**clear ip fpm flows**	Clears the IPv4 flow table of the flow platform.
Ruijie(config)# **ip fpm flow alert interval** *seconds*	Configures the IPv4 flow overflow alarm interval of the flow platform.
Ruijie(config)# **ip fpm flow alert threshold** *percent-value*	Configures the IPv4 flow overflow alarm threshold of the flow platform.
Ruijie(config)#**ip fpm flow max-entries** *flow-number*	Configures the maximum number of flow entries in the IPv4 flow table.
Ruijie(config)#**ip fpm frq** *queue-number*	Configures the number of concurrent IPv4 fragment reassembly queues.
Ruijie(config)**ip fpm session filter** *acl-number*	Protects the IPv4 flow table against attacks.
Ruijie#**show ip fpm counters**	Displays IPv4 packet counters of the flow platform.
Ruijie#**show ip fpm flows** [**filter** *protocol-number src-ip src-mask dst-ip dst-mask*]	Displays the IPv4 flow table.
Ruijie#**show ip fpm statistics**	Displays IPv4 statistics of the flow platform.
Ruijie#**show ip fpm users**	Displays the number of IPv4 user connections of the flow platform.
Ruijie#**clear ipv6 fpm flows**	Clears the IPv6 flow table of the flow platform.
Ruijie#**clear ipv6 fpm statistics**	Clears IPv6 statistics of the flow platform.

Ruijie(config)**ipv6 fpm flow alert interval** *seconds*	Configures the IPv6 flow overflow alarm interval of the flow platform.
Ruijie(config)**ipv6 fpm flow alert threshold** *percent-value*	Configures the IPv6 flow overflow alarm threshold of the flow platform.
Ruijie(config)**ipv6 fpm flow max-entries** *flow-number*	Configures the maximum number of flow entries in the IPv6 flow table.
Ruijie(config)**ipv6 fpm frq** *queue-number*	Configures the number of concurrent IPv6 fragment reassembly queues.
Ruijie(config)**ipv6 fpm session filter** *acl-name*	Protects the IPv6 flow table against attacks.
Ruijie#**show ipv6 fpm statistics**	Displays IPv6 statistics of the flow platform.
Ruijie#**show ipv6 fpm statistics fragment**	Displays IPv6 fragment reassembly statistics of the flow platform.
Ruijie#**show ipv6 fpm flows** [**filter** *protocol-number src-ip dst-ip*]	Displays the IPv6 flow table.

Configuration Examples

Example of IPv4 Configuration of the Flow Platform

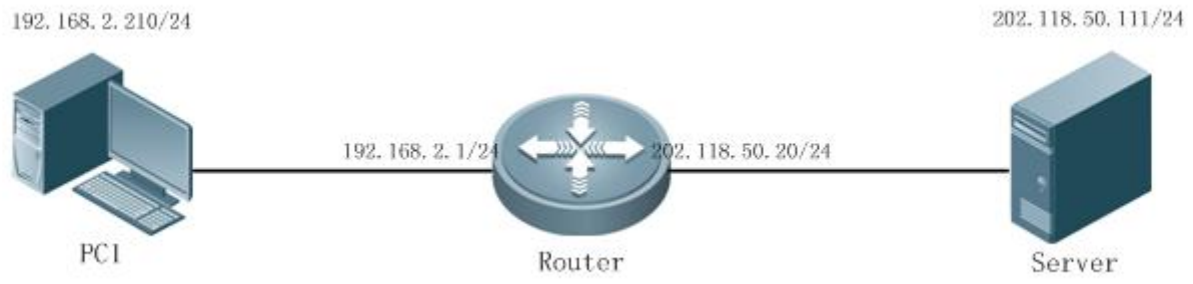
Networking Requirements

As shown in Figure 1-1, a router is connected to a PC and a server. Configure the ACL function on the router to implement the following functions:

- Enable the flow platform on the router.
- Adjust the maximum number of flow entries in the IPv4 flow table of the flow platform on the router.
- Protect the IPv4 flow table on the router by allowing flow establishment for only IP packets from the network segment 192.168.2.0/24 instead of packets from other network segments.
- Configure the IPv4 flow overflow alarm interval as 30s and the IPv4 flow overflow alarm threshold as 80% on the router.

Networking Topology

Figure 1-1 Example of IPv4 Configuration of the Flow Platform



Configuration Tips

None.

Configuration Steps

- Apply the ACL function to the loopback interface on the router to enable the flow platform.

```
Ruijie # configure terminal

Ruijie(config)# ip access-list standard 1

Ruijie(config-std-nacl)# permit any

Ruijie(config-std-nacl)# exit

Ruijie(config)# interface Loopback 0

Ruijie(config-if-Loopback 0)# ip access-group 1 in

Ruijie(config-if-Loopback 0)# exit
```

- Configure the maximum number of flow entries in the IPv4 flow table as 100,000.

```
Ruijie(config)#  ip fpm flow max-entries 100000

FPM subsystem is reinitializing...

Ruijie(config)#*Oct  6 17:35:21: %FPM-5-RESTARTED: The device IPv4 flow max-entries changed.
```

- Configure an ACL numbered 2 on the router so as to protect the IPv4 flow table against attacks.

```
Ruijie(config)#  ip access-list standard 2

Ruijie(config-std-nacl)# permit 192.168.2.0 0.0.0.255
Ruijie(config-std-nacl)# exit

Ruijie(config)# ip fpm session filter 2
```

- Configure the IPv4 flow overflow alarm interval as 30s and the IPv4 flow overflow alarm threshold as 80% on the router.

```
Ruijie(config)# ip fpm flow alert interval 30

Ruijie(config)# ip fpm flow alert threshold 80
```

Verification

Run the **ping 202.118.50.111 ntimes 1** command on PC 1.

```
C:\>ping 192.168.50.1 -n 1

Pinging 202.118.50.111 with 32 bytes of data:

Reply from 202.118.50.111: bytes=32 time=1ms TTL=64

Ping statistics for 202.118.50.111:

   Packets: Sent = 1, Received = 1, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:

   Minimum = 1ms, Maximum = 1ms, Average = 1ms
```

Run the **show ip fpm flows** command on the router. A flow entry is generated during the ping operation.

Construct a packet with a source IP address in another network segment on PC 1, and send the packet to 202.118.50.111. No corresponding flow entry can be seen in the flow table on the router.

Example of IPv6 Configuration of the Flow Platform

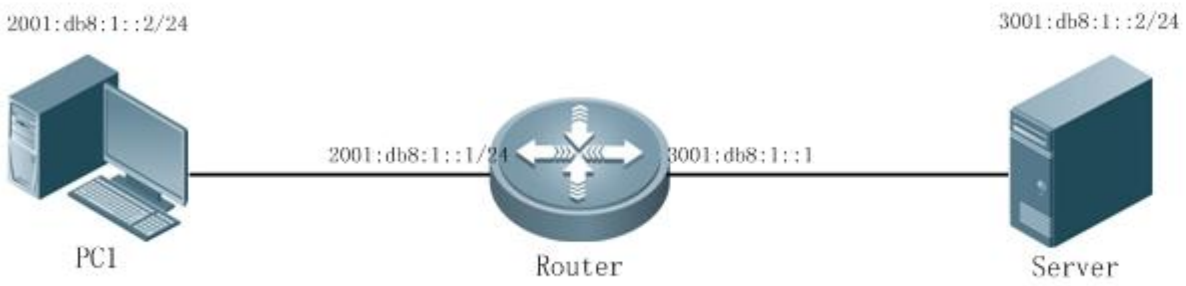
Networking Requirements

As shown in Figure 1-2, a router is connected to a PC and a server. Configure the ACL function on the router to implement the following functions:

- Enable the flow platform on the router.
- Adjust the maximum number of flow entries in the IPv6 flow table of the flow platform on the router.
- Protect the IPv6 flow table on the router by allowing flow establishment for only IP packets from the network segment 2001:db8:1::2/64 instead of packets from other network segments.
- Configure the IPv6 flow overflow alarm interval as 30s and the IPv6 flow overflow alarm threshold as 80% on the router.

Networking Topology

Figure 2-1 Example of IPv6 Configuration of the Flow Platform



Configuration Tips

None

Configuration Steps

2) Apply the ACL function to the loopback interface on the router to enable the flow platform.

```
Ruijie # configure terminal

Ruijie(config)# ip access-list standard 1

Ruijie(config-std-nacl)# permit any

Ruijie(config-std-nacl)# exit

Ruijie(config)# interface Loopback 0

Ruijie(config-if-Loopback 0)# ip access-group 1 in

Ruijie(config-if-Loopback 0)# exit
```

aa. Configure the maximum number of flow entries in the IPv6 flow table as 100,000.

```
Ruijie(config)# ipv6 fpm flow max-entries 100000

FPM subsystem is reinitializing...

Ruijie(config)#*Oct  6 17:35:21: %FPM-5-RESTARTED: The device IPv6 flow max-entries changed.
```

bb. Configure an IPv6 ACL named "virus_filter" on the router to protect the IPv6 flow table against attacks.

```
Ruijie(config)# ipv6 access-list virus_filter
Ruijie(config-ipv6-acl)# permit ipv6 2001:db8:1::/64 any

Ruijie(config-ipv6-acl)# permit icmp 2001:db8:1::/64 any

Ruijie(config)# ipv6 fpm session filter virus_filter
```

cc. Configure the IPv6 flow overflow alarm interval as 30s and the IPv6 flow overflow alarm threshold as 80% on the router.

```
Ruijie(config)# ipv6 fpm flow alert interval 30

Ruijie(config)# ipv6 fpm flow alert threshold 80
```

Verification

Run the **ping 3001:db8:1::2 ntimes 1** command on PC 1.

```
Ruijie#ping 3001:db8:1::2 ntimes 1

Sending 1, 100-byte ICMP Echoes to 3001:db8:1::2, timeout is 2 seconds:

  < press Ctrl+C to break >

!

Success rate is 100 percent (1/1), round-trip min/avg/max = 10/10/10 ms
```

Run the **show ipv6 fpm flows** command on the router. A flow entry is generated during the ping operation.

Construct a packet with a source IP address in another network segment on PC 1, and send the packet to 3001:db8:1::2. No corresponding flow entry can be seen in the flow table on the router.

Interface Configuration

Configuring Interfaces

Interface Overview

RSR series devices supports two types of interfaces: physical interfaces and logical interfaces. A physical interface is an interface that has a corresponding physical hardware port on the device.

A logical interface is an interface that has no corresponding physical hardware port on the device. A logical interface can be associated with a physical interface or independent of a physical interface. For example, dialer interfaces, Null interfaces, loopback interfaces, and sub-interfaces are logical interfaces. For network protocols, physical interfaces and logical interfaces are treated in the same way.

Ruijie series devices support the following types of interfaces:

Interface Type	Interface Configuration Name	Standard Compliance
Fast Ethernet interface	GigabitEthernet Aggregateport	IEEE802.3, RFC894
E1/CE1 port	E1	G.775, G.704, G.706, G.732
Dialer interface	dialer	—
Loopback interface	Loopback	—
NULL interface	NULL	—
Sub-interface	Serial0.1 (example)	—

Configuring Common Interfaces

Entering Interface Configuration Mode

Before you configure an interface, use the following commands to first enter global configuration mode and then enter interface configuration mode.

Command	Function
Ruijie(config)# **interface** *interface-type interface-number*	Creates an interface and enters interface configuration mode.
Ruijie(config)# **no interface** *interface-type interface-number*	Deletes the specified interface.

For example, to enter port 0 of slot 0 of the Fast Ethernet, perform the following steps:

```
Ruijie# config terminal
Ruijie(config)# interface FastEthernet 0/0
```

Note For the names of various interface types, see the interface type table above.

Note For a Ethernet port, the interface number consists of a slot number and a port number.

Note For E1/CE1 interfaces, the interface number consists of a slot number, port number and a channel number. For example, the first channel group of the third port of the E1/CE1 module in slot 2 is represented as serial 2/3:1.

Note Some interfaces have features. When you create and enter such an interface, you can specify its features. For example, to enter the frame relay point-to-point sub-interface, use the interface serial 1/0.1 point-to-point command.

Configuring IP Addresses

Except the NULL interface, every interface has its own IP address. Therefore, you must configure an IP address for an interface before you use the interface.

Use the following commands to configure or delete the IP address of an interface.

Command	**Function**
Ruijie(config-if)# **ip address** *ip-address ip-mask*	Configures the IP address of an interface.
Ruijie(config-if)# **no ip address**	Deletes the IP address of the interface.

For the details about IP address configuration, see the related chapter in the *IP Address and Service Configuration Guide*.

Configuring Interface Descriptions

Interface descriptions are used to identify interfaces.

Use the following commands to configure an interface description in interface configuration mode.

Command	**Function**
Ruijie(config-if)# description *interface-description*	Describes the purpose of the specified interface, which is a string of up to 80 characters.
Ruijie(config-if)# no description	Deletes the description of the interface.

Setting the Maximum Transmit Unit (MTU)

The MTU is a feature of IP packets. It ranges from 64 to 65535 bytes depending on the interface type. Use the following commands to set the MTU.

Command	Function
Ruijie(config-if)# **mtu** *bytes*	Configures the MTU size.
Ruijie(config-if)# **no mtu**	Restores the default value of the MTU.

Configuring Bandwidth

The bandwidth is used for some routing protocols such as OSPF to calculate the route metric and for RSVP to calculate the reserved bandwidth. Modifying the interface bandwidth will not affect the data transmission rate of the physical interface.

Use the following commands to configure the bandwidth of an interface in interface configuration mode.

Command	Function
Ruijie(config-if)# **bandwidth** *kilobits*	Configures the bandwidth
Ruijie(config-if)# **no bandwidth**	Removes the settings of the bandwidth.

Configuring the Queue Size on an Interface

The Hold-Queue allows you to modify the size of the incoming and outgoing queues of each interface. This parameter can adjust the ability of the interface to process burst data.

Use the following commands to configure the size of the incoming and outgoing queues of an interface in interface configuration mode.

Command	Function
Ruijie(config-if)# **Hold-Queue l***ength* {**in** | **out**}	Configures the size of the incoming and outgoing queues of an interface.
Ruijie(config-if)# **no Hold-Queue**	Removes the settings.

However, adjusting the size of the incoming and outgoing queues of the interface will increase network delay, which may affect real-time interactive applications. Unless necessary, it is not recommended that you modify the sizes of queues on an interface.

Monitoring and Maintaining Interfaces

Monitoring the Status of the Interface and Controller

Command	Function
Ruijie# **show interface** [**FastEthernet** | …]	Shows the transmission feature and protocol feature of the interface.

The states of the interfaces encapsulated with different link layer protocols may have some different contents when shown. See the *WAN Protocol Configuration Guide* for the monitoring and maintaining commands of various WAN protocols. For example, when the interface is encapsulated with frame relay, the **show interface serial** *2/0* command will show interface information about frame relay.

Clearing and Resetting Interface Counters

The statistics on the interface vary with the change of communication. Sometimes, to avoid the interference of communication statistics, you need to clear the statistics of the interface, so that the current statistics can faithfully reflect the current communication state of the interface.

Command	Function
Ruijie# Clear counters [**FastEthernet**] | …	Clears the communication statistics counter of the interface shown by using the show interface command to 0.
Ruijie# **Clear interface** [**FastEthernet**] | …	Clears all state values of an interface.

For example, before you use the **clear counter** command, use the **show interface serial *1/0*** command to show information about the interface:

```
Ruijie# show interface serial 1/0
serial 1/0 is DOWN  , line protocol is DOWN
Hardware is Infineon DSCC4 PEB20534 H-10 serial
Interface address is: 192.168.10.10/24
MTU 1500 bytes, BW 2000 Kbit
Encapsulation protocol is PPP, loopback not set
Keepalive interval is 10 sec , set
Carrier delay is 2 sec
RXload is 1 ,Txload is 1
LCP Closed
Closed: ipcp
Queueing strategy: WFQ
5 minutes input rate 0 bits/sec, 0 packets/sec
5 minutes output rate 0 bits/sec, 0 packets/sec
1425 packets input, 22800 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 abort
1425 packets output, 22800 bytes, 0 underruns
0 output errors, 0 collisions, 2 interface resets
6 carrier transitions
V35 DTE cable
DCD=down  DSR=down  DTR=up  RTS=up  CTS=down
Ruijie# clear counter serial 1/0
```

Then, use the **show interface** command to show information about the interface:

```
Ruijie# show interface serial 1/0
serial 1/0 is DOWN  , line protocol is DOWN
Hardware is Infineon DSCC4 PEB20534 H-10 serial
Interface address is: 192.168.10.10/24
MTU 1500 bytes, BW 2000 Kbit
Encapsulation protocol is PPP, loopback not set
```

```
Keepalive interval is 10 sec , set
Carrier delay is 2 sec
RXload is 1 ,Txload is 1
LCP Closed
Closed: ipcp
Queueing strategy: WFQ
5 minutes input rate 0 bits/sec, 0 packets/sec
5 minutes output rate 0 bits/sec, 0 packets/sec
0 packets input, 0 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 abort
0 packets output, 0 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets
0 carrier transitions
V35 DTE cable
DCD=down  DSR=down  DTR=up  RTS=up  CTS=down
```

Shutting Down and Restarting the Interface

When necessary, the interface must be shut down, for example, when you replace the cables on the interface and then restart the interface. The **shutdown** command allows you to shut down an interface, while the **no shutdown** command allows you to restart the interface.

Interface Configuration Example

Enabling an Interface

The following example configures a serial port to run PPP.

```
Ruijie(config)# interface serial 1/0
Ruijie(config-if)# encapsulation ppp
```

Configuring an Interface Description

The following example configures the function description of an Ethernet interface.

```
Ruijie(config)# interface FastEthernet 0/0
Ruijie(config-if)# description Gateway_of_translation
Ruijie(config-if)# ip address 192.168.12.1 255.255.255.0
```

Shuting Down an Interface

If an interface is idle, you can shut down it, as shown in the following example.

```
Ruijie(config)# Interface serial 1/0
Ruijie(config-if)# shutdown
```

Configuring Ethernet Interfaces

Configuring Ethernet Interfaces

Entering Ethernet Interface Configuration Mode

Use the following commands to enter Ethernet interface configuration mode in global configuration mode.

Command	Function
Ruijie(Config)# **interface GIgabitethernet** *interface-number*	Enters Gigabit Ethernet interface configuration mode.

Configuring the IP Address

Use the following commands to configure the IP address of an Ethernet interface in interface configuration mode.

Command	Function
Ruijie(Config-if)# **ip address** *ip-address ip-mask* [secondary]	Configures the IP address and its mask.
Ruijie(Config-if)# **no ip address** *ip-address ip-mask* [secondary]	Removes the settings.

Ethernet interfaces support multiple IP addresses, with the Secondary keyword used to identify the other IP addresses.

Configuring the MAC Address

Each Ethernet interface has a globally unique MAC address by default. If necessary, you can modify the MAC address of an Ethernet interface, provided that the MAC address is unique in the same LAN.

Use the following commands to configure the MAC address of an Ethernet interface in interface configuration mode.

Command	Function
Ruijie(Config-if)# **mac-address** *mac-address*	Configures the MAC address.
Ruijie(Config-if)# **no mac-address**	Removes the configured MAC address.

MAC address settings may affect communication inside the LAN. Therefore, it is not recommended that you configure an MAC address on your own unless really necessary.

Monitoring the Ethernet Interface Status

Use the following commands to monitor the Ethernet interface status.

Command	Function

Command	Function
Ruijie# **show interface GigabitEthernet** *interface-number*	Shows the status of a Gigabit Ethernet interface .

The following example shows the status of a Gigabit Ethernet interface:

```
Ruijie# show interface GigabitEthernet 0/0
GigabitEthernet0/0 is up, line protocol is up
Hardware is PQ3 TSEC GIGABIT ETHERNET CONTROLLER GigabitEthernet, address is 0a0b.0c0d.0e0e
(bia 0a0b.0c0d.0e0e)
Internet address is 192.168.200.1/24
MTU 1500 bytes, BW 1000000 Kbit
Encapsulation protocol is Ethernet-II, loopback not set
Keepalive interval is 10 sec, set
Carrier delay is 2 sec
RXload is 1 ,Txload is 1
Queueing strategy: FIFO
Output queue 0/40, 0 RSRps;
Input queue 0/75, 0 RSRps
5 minutes input rate 281 bits/sec, 0 packets/sec
5 minutes output rate 0 bits/sec, 0 packets/sec
82 packets input, 7947 bytes, 0 no buffer
Received 76 broadcasts, 0 runts, 0 giants
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 abort
0 packets output, 0 bytes, 0 underruns
0 output errors, 0 collisions, 3 interface resets
Link Mode: 1000M/Full-Duplex, media-type is twisted-pair.
Output flowcontrol is off;Input flowcontrol is off.
```

Clearing and Resetting the Ethernet Interface

Use the following commands to clear and reset the internal counters of an Ethernet interface.

Command	Function
Ruijie# **clear counters GigabitEthernet** *interface-number*	Clears and resets the internal counters of a Gigabit interface.

Ethernet Interface Configuration Example

The Ethernet interface configuration is very simple. The basic task is to configure the IP address of the interface. For others, you can often accept the factory defaults. For example:

```
Ruijie# config terminal
Ruijie(config)# interface fastEthernet 0/0
Ruijie(config-if)# ip address 192.168.12.1 255.255.255.0
Ruijie(config-if)# no shutdown
```

Configuring VLANs

The Virtual Local Area Network (VLAN), a logical network subdivided from a physical network, is a Layer 2 network in the ISO model. IEEE published the 802.1Q draft in 1999 to standardize VLAN implementation.

The VLAN technology allows the network administrator to split a physical LAN into different broadcast areas (or called VLANs) logically. Each VLAN consists of a group of computer workstations with the same needs. So they have the same features as the physical LAN. However, since VLANs are defined logically but not physically, the workstations in a VLAN do not need to be configured in the same physical space. In other words, these workstations do not necessarily belong to the same physical LAN network segment. The broadcast and unicast traffic in a VLAN will not be forwarded to other VLANs, which can facilitate flow control, reduce equipment investment, simplify network management, and improve network security.

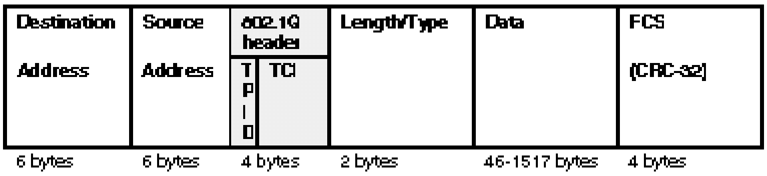
The VLAN is a protocol originated to solve the Ethernet broadcast problem and security. On the basis of Ethernet frames, a VLAN header is added and a VLAN ID is used to assign users into smaller work groups to restrict the mutual access on L2 between different workgroups. Each workgroup is a VLAN. The VLAN allows you to restrict the broadcast range and form a virtual workgroup for dynamic network management.

802.1Q

For details about 802.1Q, see the protocol specification. This section only briefly describes the 802.1Q frame format for your reference.

As shown in 0, when a host supporting 802.1Q sends a packet, it adds a 4-byte 802.1Q frame header before the source address of the old Ethernet frame header, followed by the Length or Type field of the old Ethernet frame.

Figure 2-1 Ethernet Frame with the 802.1Q Tag Header



Destination Address	Source Address	802.1Q header		Length/Type	Data	FCS (CRC-32)
		TPID	TCI			
6 bytes	6 bytes	4 bytes		2 bytes	46-1517 bytes	4 bytes

This 4-byte 802.1Q tag header consists of a 2-byte TPID (Tag Protocol Identifier, with the value of 0x8100) and a 2-byte TCI (Tag Control Information). The TPID is a new type defined by IEEE and it indicates that the packet has been added with the 802.1Q tag. 0 shows details about the 802.1Q tag header.

Figure 2-2 802.1.1Q Tag Header

TPID (Tag Protocol Identifier)		TCI		
0x81	0x00	P	C	VLAN ID

31 23 15 12 0

The information in the tag header is described as below:

VLAN Identified (VLAN ID): This is a 12-bit field, which specifies the ID of the VLAN, totally 4096. Each packet supporting 802.1Q from the host contains this field to indicate that it belongs to which VLAN.

Canonical Format Indicator (C): This bit specifies T-R encapsulation, used for compatibility with the token network, so that the data of the token network can be normally transmitted over the Ethernet without conversion.

Priority (P): It is a 12-bit field that specifies the priority of the frame. There are totally eight priorities, used to select the packets first to be sent when the device is in congestion.

On a network divided with VLANs, devices can forward data, and implement traffic control and broadcast management between VLANs.

Configuring the VLAN Encapsulation Tag

Use the following commands to configure the VLAN encapsulation tag.

Command	Function
Ruijie(config)# **interface FastEthernet** *Sub_interface_number*	Enters or creates an 802.1Q encapsulated sub-interface.
Ruijie((config-subif))# **encapsulation dot1Q** *VlanID*	Encapsulates 802.1Q and specifies a VLAN ID, where the VLANID must be the same as one of the VLAN IDs on the device.

The VLAN ID is in the range from 1 to 4094.

Note Be sure to use the proper port of RSR series routers to connect the VLAN native-enabled devices of other vendors.

Assigning IP Addresses to Interfaces

After you encapsulate the VLAN tag, you must specify the IP address for the Ethernet sub-interface encapsulated with the VLAN tag.

Use the following commands to configure the IP address of an interface in sub-interface configuration mode.

Command	Function
Ruijie((config-subif))# **ip address** *ip-address mask*	Specifies the IP address of an interface.

The IP address of the Ethernet interface encapsulated with 802.1Q is usually the gateway through which the hosts in the VLAN are connected with the hosts in another VLAN.

Debugging and Monitoring VLANs

Use the following commands to debug and monitor the VLAN in privileged user mode.

Command	Function
Ruijie# **debug vlan**	Enables VLAN encapsulation event debugging .

Command	Function
Ruijie# **no debug vlan**	Disables VLAN encapsulation event debugging.
Ruijie# **show vlans** [*VlanID*]	Shows VLAN statistics.
Ruijie# **clear vlan** [*VlanID*]	Clears VLAN statistics.

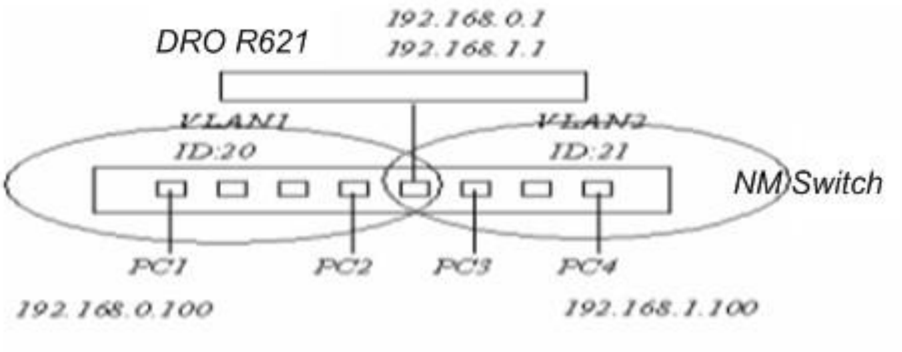
VLAN Configuration Example

Configuration Requirements

Set two VLANs on the VLAN-supported network mangement device:

1. One VLAN is connected to hosts in the 192.168.0.0/24 network segment, and the other VLAN is connected to hosts in the 192.168.1.0/24 network segment.

2. The Ethernet port of the device is encapsulated with 802.1Q to serve as a gateway for mutual communication between hosts in the two VLANs, as shown in the following figure:



Equipment Configuration

Equipment configuration:

```
hostname "Ruijie"
!
interface FastEthernet0/0
no ip address
!
interface FastEthernet0/0.20
encapsulation dot1Q 20                  //Configure 802.1Q encapsulation, with the VlANID as
20
ip addr 192.168.0.1 255.255.255.0       //Configure the IP address of the sub-interface
!
interface FastEthernet0/0.21
encapsulation dot1Q 21                  //Configure 802.1Q encapsulation, with the VlANID as
21
ip addr 192.168.1.1 255.255.255.0       //Configure the IP address of the sub-interface
```

```
!
interface FastEthernet0/1
no ip address
shutdown
!
```

End

Configuring WAN Interfaces

Configuring CE1 Interfaces

CE1 Interface Overview

The CE1 interface works in two modes: E1 and CE1.

E1 working mode: timeslot division does not apply. It is equivalent to an interface whose data bandwidth is 2.048 Mbit/s. In this mode, the CE1 interface supports multiple encapsulation protocols, including PPP, Frame Relay, X.25, HDLC and LAPB.

CE1 working mode: It is the channelized timeslot working mode, where the 2.048 Mbit/s bandwidth is divided into 8K frames per second. Each CE1 frame has 256 bits, and is further divided into 32 timeslots, so each timeslot has 8 bits. The 32 timeslots are identified as 0-31. Among them, timeslot 0 is used to transmit CE1 frame synchronization signals, and timeslots 1-31 are used to transmit data. Accordingly, you can configure up to 31 channel groups, whose numbers are 0-30. Each channel group is effective when one to 31 timeslots are configured. A timeslot may be idle or assigned to a channel group, but cannot be assigned to multiple channel groups. A channel group assigned with timeslots is a logical interface. Its bandwidth is the number of occupied timeslots multiplexed by 64 kbit/s. It supports multiple encapsulation protocols, including PPP, Frame Relay, X.25, HDLC and LAPB.

Configuring CE1 Interfaces

Entering CE1 Interface Configuration Mode

Command	Function
Ruijie(config)# **controller e1** *slot-number / port-number*	Enters CE1 interface configuration mode.

Setting the Timeslot Allocation Mode of the CE1 Interface

Use the **channel-group** command to set the timeslot configuration mode (provided that the interface works in CE1 mode) in CE1 interface configuration mode.

Command	Function
Ruijie(config-controller)# **channel-group** *chan-num* **timeslots** *timeslot-range*	Allocates the specified timeslot range (timeslot numbers within 1-31, not necessarily continuous) to the specified channel group (channel group numbers within 0-30).
Ruijie(config-controller)# **no channel-group** *chan-num*	Removes the configuration.

After you use the above command to allocate the timeslots to a channel group, the system generates a logical interface, for example:

```
Ruijie(config)# controller e1 1/1/0
Ruijie(config-controller)# channel-group 0 timeslots 1-2 4
```

The logical interface serial 1/1/0:0 is generated.

Setting the Channel-Group Working Parameters

Command	Function
Ruijie(config)# **interface serial** *slot-number / port-number:chan-num*	Enters the logical interface configuration mode of the specified channel group.

Setting the Sync Clock Source

When two CE1 interfaces are directly connected, you must specify the synchronous clock source. When one CE1 interface provides the synchronous clock, the other CE1 interface obtains the clock from the data reception line as the synchronous clock for the reception/transmission of the local data. When the CE1 interface is connected with the device, usually the switch provides the synchronous clock.

Command	Function
Ruijie(config-controller)# **clock source** { **line** | **internal** }	Sets the sync clock source of the CE1 interface.
Ruijie(config-controller)# **no clock source**	Restores the default sync clock source of the CE1 interface.

The default sync clock source of the CE1 is *line*.

Setting the Frame Check Mode

When the CE1 interface works in CE1 mode, the 4-bit CRC of the physical frames is supported.

Command	Function
Ruijie(config-controller)#**framing** { **crc4** | **no-crc4** }	Sets the frame check mode of the CE1 interface.
Ruijie(config-controller)#**no framing**	Restores the default frame check mode of the CE1 interface.

The default frame check mode of the CE1 interface is CRC4.

Setting the Line Coding/Decoding Mode

The CE1 interface supports two line coding/decoding modes:

- AMI mode
- HDB3 mode

Command	Function

Command	Function
Ruijie(config-controller)# **linecode** { **ami** | **hdb3** }	Sets the line coding/decoding mode of the CE1 interface.
Ruijie(config-controller)# **no linecode**	Restores the default line coding/decoding mode of the CE1 interface.

The default line coding/decoding mode of the CE1 interface is HDB3.

Enabling or Disabling Local Loopback

Command	Function

Ruijie(config-controller)# **loopback** { **local** }
or
Ruijie(config-controller)# **loopback { local | network** }
	Enables local loopback of the CE1 interface.

Ruijie(config-controller)# **no loopback**	Disables local loopback of the CE1 interface.

By default, the local loopback of the CE1 interface is disabled.

Setting the E1 or CE1 Working Mode

The CE1 interface can work in E1 or CE1 working mode.

Command	Function
Ruijie(config-controller)# **using** { **ce1** | **e1** }	Set sthe CE1 or E1 working mode.
Ruijie(config-controller)# **no using**	Restores the default working mode.

The default working mode is CE1.

Setting the E1 Working Parameters

When the CE1 interface works in E1 mode, the system automatically generates an E1 logical interface *serial slot-number/port-number:0*.

Command	Function
Ruijie(config)# **interface serial** *slot-number/port-number: 0*	Enters the configuration mode of the specified E1 logical interface.

Monitoring and Maintaining CE1 Interfaces

Ruijie products provide the **show controller e1** command for monitoring and maintaining the CE1 interface. The **show interface** command allows you to monitor and maintain the channel group and E1 logical interface.

Command	Function
Ruijie# **show controller e1** [*slot-number/port-number*]	Shows the related status information about the (specified) CE1 interface.

Command	Function
Ruijie# **show interface serial** *slot-number/port-number:chan-num*	Shows the status information about the specified channel-group logical interface.
Ruijie# **show interface serial** *slot-number/port-number:0*	Shows the status information about the specified E1 logical interface.

Example 1: Show the status information about all CE1 interfaces.

```
Ruijie# show controller e1
E1 1/1/0 is down.
Applique type is Channelized E1 - balanced
Receiver has no alarms.
Framing is crc4, Line Code is hdb3, Clock Source is line
Data in current interval (514 seconds elapsed):
0 Line Code Violations, 0 Path Code Violations
0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
Total Data (last 2 fifteen minute intervals):
0 Line Code Violations, 0 Path Code Violations
0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
E1 1/1/1 is down.
Applique type is Channelized E1 - balanced
Receiver has no alarms.
Framing is crc4, Line Code is hdb3, Clock Source is line
Data in current interval (514 seconds elapsed):
0 Line Code Violations, 0 Path Code Violations
0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
Total Data (last 2 fifteen minute intervals):
0 Line Code Violations, 0 Path Code Violations
0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
Applique type is Channelized E1 - balanced
Receiver has no alarms.
Framing is crc4, Line Code is hdb3, Clock Source is line
Data in current interval (514 seconds elapsed):
0 Line Code Violations, 0 Path Code Violations
0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
Total Data (last 2 fifteen minute intervals):
0 Line Code Violations, 0 Path Code Violations
0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
```

```
E1 1/1/3 is down.
Applique type is Channelized E1 - balanced
Receiver has no alarms.
Framing is crc4, Line Code is hdb3, Clock Source is line
Data in current interval (514 seconds elapsed):
0 Line Code Violations, 0 Path Code Violations
0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
Total Data (last 2 fifteen minute intervals):
0 Line Code Violations, 0 Path Code Violations
0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
```

Example 2: Show the status information about the specified CE1 interface.

```
Ruijie# show controller e1 1/0
E1 1/0 is down.
Applique type is Channelized E1 - balanced
Receiver has no alarms.
Framing is crc4, Line Code is hdb3, Clock Source is line
Data in current interval (615 seconds elapsed):
0 Line Code Violations, 0 Path Code Violations
0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
Data in Interval 1:
0 Line Code Violations, 0 Path Code Violations
0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
Data in Interval 2:
0 Line Code Violations, 0 Path Code Violations
0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
Total Data (last 2 fifteen minute intervals):
0 Line Code Violations, 0 Path Code Violations
0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
```

The above infomration includes the physical status, working mode, frame check mode, line coding/decoding mode and sync clock source of the specified CE1 interface. It also includes statistics in the past 24 hours and every 15 minutes, as well as overall statistics.

Example 3: Show the status information about the specified channel-group logical interface.

```
Ruijie# show interface serial 1/1/0:0
serial 1/1/0:0 is DOWN  , line protocol is DOWN
Hardware is  serial
Interface address is: no ip address
```

```
MTU 1500 bytes, BW 192 Kbit
Encapsulation protocol is HDLC, loopback not set
Keepalive interval is 10 sec , set
Carrier delay is 2 sec
RXload is 1 ,Txload is 1
Queueing strategy: WFQ
5 minutes input rate 0 bits/sec, 0 packets/sec
5 minutes output rate 0 bits/sec, 0 packets/sec
0 packets input, 0 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 abort
0 packets output, 0 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets
Timeslot(s) Used:1-2,4, Transmitter delay is 0 flags
```

Example 4: Show the status information about the specified E1 logical interface.

```
Ruijie# show interface serial 1/1/1:0
serial 1/1/1:0 is DOWN  , line protocol is DOWN
Hardware is  serial
Interface address is: no ip address
MTU 1500 bytes, BW 2000 Kbit
Encapsulation protocol is HDLC, loopback not set
Keepalive interval is 10 sec , set
Carrier delay is 2 sec
RXload is 1 ,Txload is 1
Queueing strategy: WFQ
5 minutes input rate 0 bits/sec, 0 packets/sec
5 minutes output rate 0 bits/sec, 0 packets/sec
0 packets input, 0 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 abort
0 packets output, 0 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets
Timeslot(s) Used:0-31, Transmitter delay is 0 flags
```

The above information includes the physical status, link status, encapsulation mode, received and sent packet statistics, and timeslot occupation of the channel-group or E1 logical interface.

Configuration Examples

CE1 Configuration Examples

The following example configures the E1 interface as channelized E1 (CE1) and divides two channels: channel 1 and channel 8, where channel 1 occupies timeslots 1, 2, 3, 7, and 19, while channel 8 occupies timeslots 8, 9, 10, 11, 20, and 30. The line clock (default) is used and the line coding mode is HDB3 (default). The specific configuration is shown as follows:

```
Ruijie(config)# controller E1 1/2/0
Ruijie(config)# using ce1          //Default configuration; configured only when using e1 is currently used
Ruijie(config-controller)# channel-group 1 timeslots 1-3,7,19
Ruijie(config-controller)# channel-group 8 timeslots 8-11,20,30
```

After configuration, the following logical interfaces are created:

```
interface serial 1/2/0:1
interface serial 1/2/0:8
```

E1 Configuration Examples

To configure the E1 interface as a pure E1 interface which is not divided into timeslots, first clear the CE1 configuration, that is, clear the 31 timeslots, and then set the interface to the pure E1 working mode. For example, the configuration operations are as below according to 3.3.5.1:

```
Ruijie(config)# controller E1 1/2/0
Ruijie(config-controller)# no channel-group 1
Ruijie(config-controller)# no channel-group 8
Ruijie(config-controller)# using e1
```

If no CE1 channel is configured before, the configuration operation is as below:

```
Ruijie(config)# controller E1 1/2/0
Ruijie(config-controller)# using e1
```

After configuration, the system generates the following logical interface:

```
interface serial 1/2/0:0
```

Troubleshooting CE1 Interfaces

The CE1 interface is often used to provide the service of access to the WAN, and its logical interface is usually encapsulated with the appropriate link layer protocol. After you verify that the physical interface and the line connection are in good connection, if the CE1 logical interface link is still interrupted, you need to debug the configuration of the logical interface and the operation of the link layer protocol. For the specific procedure, see the related link layer protocol configuration section. When the CE1 logical interface is interrupted, you must first check the following Aspects:

1. Make sure that the CE1 interfaces connected have the same working mode, for example, E1 mode or CE1 mode.

2. If both ends work in CE1 mode, ensure that the channel groups at both ends are configured the same.

3. Make sure that the two CE1 interfaces connected have the same line coding/decoding and frame check modes configured.

4. Check the sync clock sources of the CE1 interfaces at both ends. One end must provide the sync clock, and the other end must obtain the sync clock signals from the data line.

5. Make sure the logical interfaces at both ends use the same link encapsulation protocol.

6. Observe if the physical status of the CE1 interface is **UP**. If not, it means that the device at the other end is shut down or the cable connection is abnormal. Please use appropriate substitutes to locate the problem.

Configuring Aggregate Port

This chapter describes how to configure an aggregate port on a Ruijie router.

Overview

Aggregate Port

An aggregate port (AP) is a logical link formed by binding multiple physical links. The AP function provided by Ruijie routers complies with IEEE802.3ad and can be used to expand link bandwidth, thereby improving connection reliability.

The AP function supports load balancing, and therefore load can be evenly distributed to member links. The AP function also implements link backup. When a member link is disconnected, the system automatically distributes load of this link to other available member links. Broadcast or multicast packets received by a member link are not forwarded to other member links.

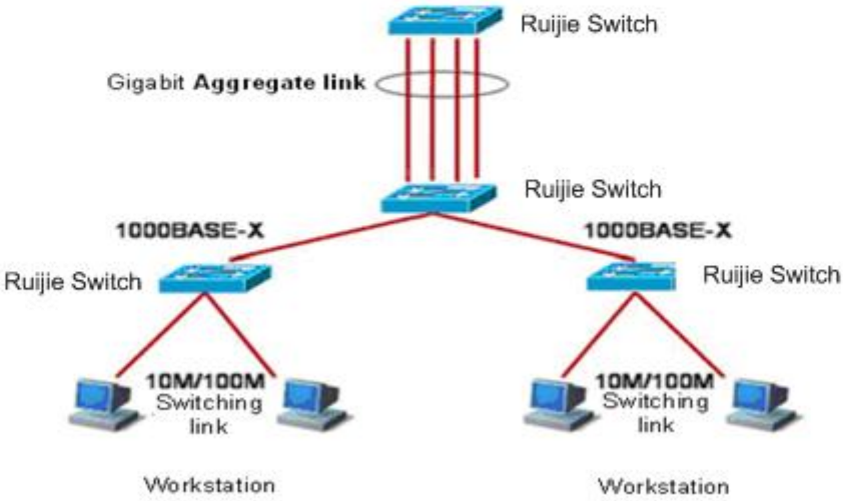


Figure 1 Typical AP configuration on a router

Each AP contains up to 8 members. The following table lists the maximum AP quantity.

Product Series	Maximum AP Quantity
RSR20-X-28	6

Load Balancing

An AP enables even load distribution to its member links based on source IP addresses, the destination IP addresses, a combination of source and destination IP addresses, or a combination of source IP addresses and destination port numbers. You can run the **aggregateport load-balance** command in interface configuration mode to set a load distribution mode.

In the load balancing mode based on source IP addresses, packets are distributed to different member links according to source IP addresses. Packets with different source IP addresses are distributed to different member links; packets with a same source IP address are forwarded along a same member link.

In the load balancing mode based on the destination IP addresses, packets are distributed to different member links according to destination IP addresses. Packets with different destination IP addresses are distributed to different member links; packets with a same destination IP address are forwarded along a same member link.

In the load balancing mode based on source and destination IP addresses, packets are distributed to different member links according to the combination of source and destination IP addresses. Packets with different source and destination IP addresses are distributed to different member links; packets with same source and destination IP addresses are forwarded along a same member link. The Layer-3 AP uses this mode by default.

In the load balancing mode based on source IP addresses and destination port numbers, packets are distributed to different member links according to the combination of source IP addresses and destination port numbers. Packets with different source IP addresses and destination port numbers are distributed to different member links; packets with a same source IP address and destination port number are forwarded along a same member link.

All the abovementioned balancing modes are applicable to L2 and L3 APs.

Balancing Mode	**RSR20-X-28**
Source MAC-based	√
Destination MAC-based	√
Source+Destination MAC-based	√
Destination IP-based	×
Source IP-based	×
Source+Destination IP-based	×

A load distribution mode is set based on the network environment, to ensure that load can be evenly distributed to different links, so that network bandwidth can be fully utilized.

In Figure 2, PC1 and PC2 communicate with PC4 to PC6 over the AP. R1 serves as a gateway for all of PC4 to PC6. Packets from an external network (PC1 and PC2) are sent via R2 take the IP addresses of PC1 and PC2 as the source IP addresses. To distribute communication load between R2 and other hosts to other links, perform load balancing based on the source IP addresses. On R1, set the load balancing mode based on destination IP addresses.

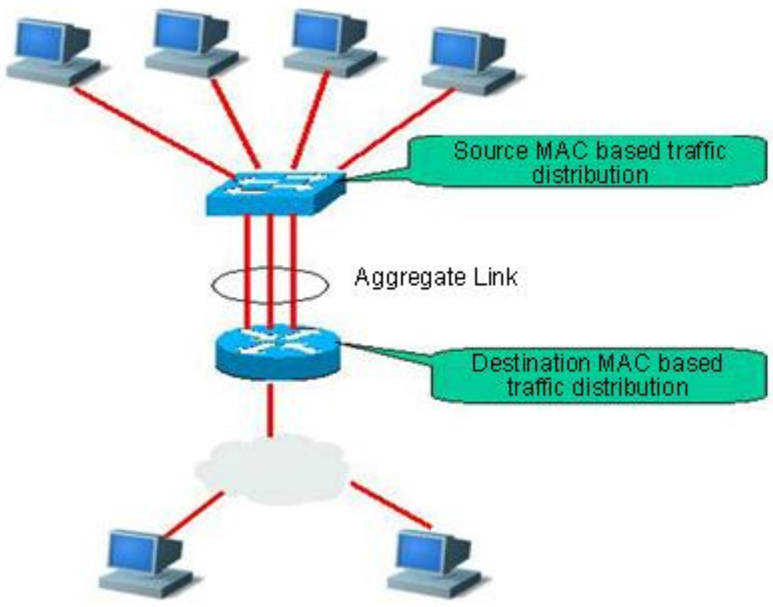

Source MAC based traffic distribution
Aggregate Link
Destination MAC based traffic distribution

Figure 2 AP load balancing

Aggregate Port Configuration

Default Configuration

The following table lists the default configuration of the AP.

Feature	Default Value
Layer-3 AP	N/A
Layer-3 AP	N/A ☑ Layer-3 AP is not supported on RSR20-X series routers.
Load balancing	**src-ip**
LinkTrap	Disabled

Configuration Guide

Port rates of AP members must be consistent.

The Layer-3 AP supports Layer-3 ports only, and therefore layer-2 ports cannot be added to the Layer-3 AP.

Do not set a port security function for the AP.

If a port is added to an AP, properties of the port are replaced by those of the AP.

Caution When a port is added to an AP, you cannot perform any configuration on the port before removing the port from the AP.

Configuring Aggregate Port

In interface configuration mode, perform the following step to add a port to an AP.

Command	Description
Ruijie(config-if-range)# **port-group** *port-group-number*	Adds a port to an AP. If the AP does not exist, create the AP at the same time.

Run the **no port-group** command in interface configuration mode to remove a physical port from the AP.

The following example is to configure Layer-2 Ethernet port 0/1 as a member of Layer-2 AP 5.

```
Ruijie# configure terminal
Ruijie(config)# interface range gigabitEthernet 0/1
Ruijie(config-if-range)# port-group 5
Ruijie(config-if-range)# end
```

You can run the Ruijie(config)# **interface aggregateport** *n* command (n indicates the AP number) in global configuration mode to create AP *n* (supposing that AP *n* does not exist).

Caution After adding an ordinary port to an AP, when the port exits from the AP again, the previous configuration of the port will be restored to the default one. Different functions deal with the previous configuration of the AP member port differently. Therefore, it is recommended to view and confirm the port configuration after exiting from the AP.

Configuring Layer-3 Aggregate Port

By default, an AP is a Layer-3 AP.

The following example is to configure a Layer-3 AP (AP 3) and set the IP address of AP 3 to 192.168.1.1.

```
Ruijie# configure terminal
Ruijie(config)# interface aggretegateport 3
Ruijie(config-if)# ip address 192.168.1.1 255.255.255.0
Ruijie(config-if)# end
```

Caution When you run the **port-group** command to add a Layer-3 port to an AP, the operation fails if the AP does not exist. You must create the Layer-3 AP first before adding the Layer-3 port to the AP.

After the Layer-3 AP is created, you can add members to the AP. The following example is to add gigabit Ethernet 0/1-3 ports to Layer-3 AP 2.

```
Ruijie# configure terminal
Ruijie(config)# interface range gigabitEthernet 0/1-3
Ruijie(config-if)# port-group 2
```

Configuring Load Balancing for Aggregate Port

Perform the following step in global configuration mode to configure a load balancing mode for an AP.

Command	Description
Ruijie(config)# **aggregateport load-balance** {**dst-mac | src-mac | src-dst-mac | dst-ip | src-ip | src-dst-ip** }	Selects one of the following load balancing modes for the AP: **src-dst-mac**: Distributes load according to source and destination MAC addresses. **dst-mac**: Distributes load according to destination MAC addresses. **src-mac**: Distributes load according to source MAC addresses. **src-dst-ip**: Distributes load according to source and destination IP addresses. **dst-ip**: Distributes load according to destination IP addresses. **src-ip**: Distributes load according to source IP addresses.

To restore the default load balancing mode of an AP, run the **no aggregateport load-balance** command in global configuration mode.

Configuring LinkTrap for Aggregate Port

Perform the following step in global configuration mode to configure the linktrap function for the AP member port.

Command	Description
Ruijie(config)# **aggregateport member linktrap**	Enables the linktrap function for the AP member port.

To disable the linktrap function, run the **no aggregateport member linktrap** command in global configuration mode.

The following example is to enable the linktrap function for the AP member port.

```
Ruijie# configure terminal
Ruijie(config)# aggregateport member linktrap
```

Displaying Aggregate Port

Perform the following step in privileged EXEC mode to display the AP configuration.

Command	**Description**
Ruijie# **show aggregateport** [*port-number*] {**load-balance | summary**}	Displays AP configuration.

```
Ruijie# show aggregateport load-balance
Load-balance : Source MAC address
Ruijie#show aggregateport 1 summary
AggregatePort MaxPorts SwitchPort Mode   Ports
------------- -------- ---------- ------
Ag1         8       Enabled    ACCESS
```

Configuring Logical Interfaces

Unlike physical interfaces, a logical interface is an interface that can exchange data but does not exist physically or needs to be configured. Ruijie products provide the following types of logical interfaces: Loopback interfaces, NULL interfaces, dialer interfaces and sub-interfaces.

Configuring Loopback Interfaces

A loopback interface is a local interface that is completely simulated by software and is always in UP status. The packets including routing information sent to a loopback interface will be processed at the device locally. The IP address of a loopback interface can be used as the device ID of the OSPF routing protocol and for the Telnet or as the network interface for remote Telnet access. The procedure for configuring a loopback interface is similar to that for configuring an Ethernet interface, and you can take it as a virtual Ethernet interface.

Command	Function
Ruijie(config)# **interface loopback** *loopback-interface-number*	Sets a loopback interface.
Ruijie(config)# **no interface loopback** *loopback-interface-number*	Deletes a loopback interface.

After you create a loopback interface with the specified interface number by using the **interface loopback** *loopback-interface-number* command, you can configure the communication parameters of the loopback interface (such as its IP address) just like an Ethernet interface. For the specific procedure, see the section about Ethernet interface configuration.

Since a loopback interface is a virtual interface and only exists logically, you can delete it by using the **no interface loopback** *loopback-interface-number* when necessary. The system provides the **show interfaces loopback** *loopback-interface-number* command for monitoring and maintaining a loopback interface.

Command	Function
Ruijie# **show interfaces loopback** *loopback-interface-number*	Shows the status of a loopback interface.

The following example shows how to use the **show interfaces loopback** command.

```
Ruijie# show interfaces loopback 0
Loopback 0 is UP  , line protocol is UP
Hardware is  Loopback
Interface address is: no ip address
MTU 1500 bytes, BW 8000000 Kbit
Encapsulation protocol is LOOPBACK, loopback not set
Keepalive interval is 0 sec , no set
Carrier delay is 2 sec
RXload is 1 ,Txload is 1
```

```
Queueing strategy: FIFO
Output queue 0/40, 0 RSRps;
Input queue 0/75, 0 RSRps
5 minutes input rate 0 bits/sec, 0 packets/sec
5 minutes output rate 0 bits/sec, 0 packets/sec
0 packets input, 0 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 abort
0 packets output, 0 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets
```

From the above information, you can know the settings and working status of the loopback interface, for example, interface and link status, IP address settings, MTU settings, and bandwidth settings.

Configuring Null Interfaces

Ruijie products provide NULL virtual interfaces. This kind of virtual interface is only equivalent to an available system device. The NULL interface is always in UP status and will never send or receive network data unsolicitedly. Any packets sent to the NULL interface will be discarded. Any attempt to encapsulate the link layer protocol on the NULL interface will fail. Not any command can be configured on the NULL interface (excluding the **help** and **exit** commands available for every interface).

The NULL interface is more often used to filter network traffic. You can route the undesired data to the NULL interface without using any ACL, as shown in the following example.

```
Ruijie(config)#ip route 127.0.0.0 255.0.0.0 null 0
```

Use the following command to configure a NULL interface.

Command	Function
Ruijie(config)# **interface nul**l *0*	Enters NULL interface configuration mode.

By functionality, the NULL interface is a null system device, so you cannot delete it by using the **no interface null** command.

Configuring Tunnel Interfaces

Tunnel Interface Overview

Ruijie products provide tunnel interfaces to implement the tunnel function by using transmission protocols like IP to transmit packets of any protocol. As with other logical interfaces, a tunnel interface is also a virtual system interface. Instead of particularly specifying any transmission protocol or load protocol, a tunnel interface provides a standard point-to-point transmission mode. Since a tunnel implements a point-to-point transmission link, a tunnel interface must be set for every individual link.

The tunnel function implementation includes the following parts:

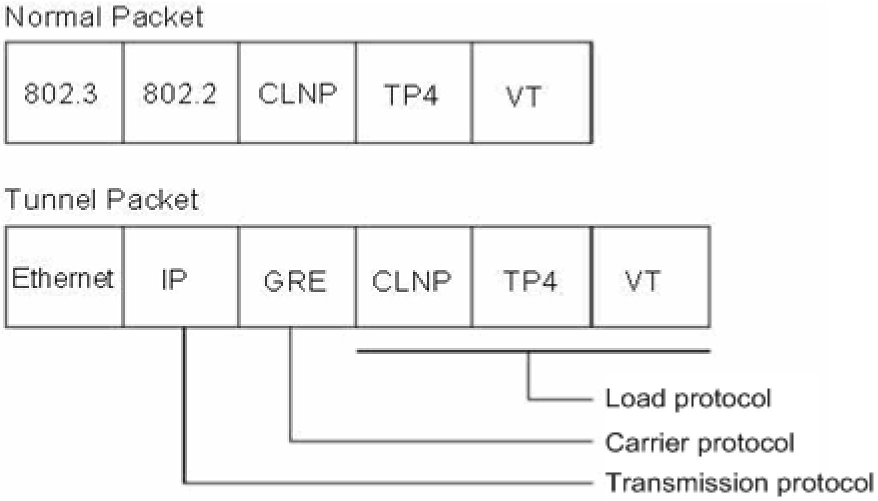
1. Load protocol: the protocol used to encapsulate data over the tunnel. Currently, Ruijie products only support the IP protocol as the load protocol on the tunnel interface.

2. Carrier protocol: the protocol used for secondary encapsulation and identification of the data to be transmitted. Ruijie products support the following encapsulation modes on the tunnel interface: GRE and IPIP.

3. Transmission protocol: network protocol of the load after the secondary encapsulation of the carrier protocol. Ruijie products use the most commonly used IP protocol as the transmission protocol.

0 shows a comparison between packets before and after transmission over the IP tunnel.

Figure 4-1 Comparison Between Packets Before and After Transmission Over the IP Tunnel



As shown in 0, IP tunnel transmission is implemented through GRE encapsulation. In fact, if two private networks running the same protocol need to communicate with each other through a heterogeneous public network, the tunnel mode can be used.

Tunnel transmission is applicable to the following cases:

Allowing the local networks running non-IP protocols to communicate with each other through a single network (IP network), since the tunnel supports multiple different load protocols.

Allowing the protocols that have limit on hops to work in a wider range, since the tunnel uses the transmission protocol (such as IP) for routing.

Allowing discrete subnets to be connected via a single network (IP network).

Allowing the VPN (virtual private network) to be enabled on the WAN.

Since the tunnel transmits data after encapsulation, this makes processing more complicated and in some cases, you need to pay attention to this change:

Since the tunnel is a point-to-point link, it is seen as a next hop in routing. However, actually its route metric may be more than one hop. In using the tunnel, you must note that the route of the tunnel link is different from the actual route.

Data is encapsulated with the transmission protocol. You must bear this in mind when you set firewalls and especially an Access Control List (ACL). At the same time, you must also note that the transmission bandwidth (like MTU) of the load protocol is smaller than the theoretical value.

The following section only describes how to configure the special attributes of the tunnel interface. For other attributes (including the IP address and other parameters, firewall and backup center), see related sections.

Configuring Tunnel Interfaces

Entering Tunnel Interface Configuration Mode

Use the following commands to create a tunnel interface and enter tunnel interface configuration mode in global configuration mode.

Command	Function
Ruijie(config)# **interface tunnel** *tunnel-number*	Enters tullel interface configuration mode.
Ruijie(config)# **no interface tunnel** *tunnel-number*	Deletes the created tunnel interface.

Same as other logical interfaces, one tunnel interface is created at the first time you enter the specified tunnel interface.

Setting the Source Address of the Tunnel Interface

For a tunnel interface, you need to configure the source address and destination address of the tunnel. To ensure the stability of the tunnel interface, the loopback address is usually used as the source address and destination address of the tunnel. Before the tunnel interface is normally working, you must verify the connectivity between the source address and the destination address.

Use the following commands to configure the source address of a tunnel interface in tunnel interface configuration mode.

Command	Function
Ruijie(config-if)# **tunnel source** { *ip-address* | *interface-name* *interface-number* }	Set sthe source address of a tunnel interface.
Ruijie(config-if)# **no tunnel source**	Cancels the source address of the tunnel interface.

The **tunnel source** command sets the source address of a tunnel interface and this source address is also the local end of the tunnel.

Setting the Destination Address of the Tunnel Interface

Use the following commands to configure the destination address of a tunnel interface in tunnel interface configuration mode.

Command	Function
Ruijie(config-if)# **tunnel destination** { *ip-address* }	Sets the destination address of a tunnel interface.
Ruijie(config-if)# **no tunnel destination**	Cancels the destination address of the tunnel interface.

The **tunnel destination** command sets the destination address of a tunnel interface and this destination address is also the remote end of the tunnel.

Caution You should not use the same source address and destination address on a device to create two or more tunnel interfaces with the same encapsulation mode.

Setting the Encapsulation Format of the Tunnel

The tunnel encapsulation format is the carrier protocol of the tunnel. By default, the tunnel interface uses the encapsulation format GRE. You can also determine the encapsulation format of the tunnel interface according to actual needs.

Command	Function
Ruijie(config-if)# **tunnel mode** { **gre ip** | **ipip** | **ipv6ip** }	Set sthe encapsulation format of the tunnel.
Ruijie(config-if)# **no tunnel mode**	Cancels the encapsulation format of the tunnel interface and restores it to the default.

Setting the Tunnel Check

In some cases, you need to enable tunnel check to ensure data integrity.

Command	Function
Ruijie(config-if)# **tunnel checksum**	Enables tunnel check.
Ruijie(config-if)# **no tunnel checksum**	Disables tunnel check.

By default, the tunnel interface does not perform check.

Setting the Key of the Tunnel Interface

By setting the key of a tunnel interface, you can ensure the security of the settings at both ends of the tunnel to some extent, and also prevent malicious external sniffing or attacks.

Command	Function
Ruijie(config-if)# **tunnel key** *key-value*	Sets the key of the tunnel interface.
Ruijie(config-if)# **no tunnel key**	Cancels the key of the tunnel interface.

The key of the tunnel interface takes effect only when the tunnel encapsulation format is GRE, so every GRE packet includes the set key of the tunnel interface.

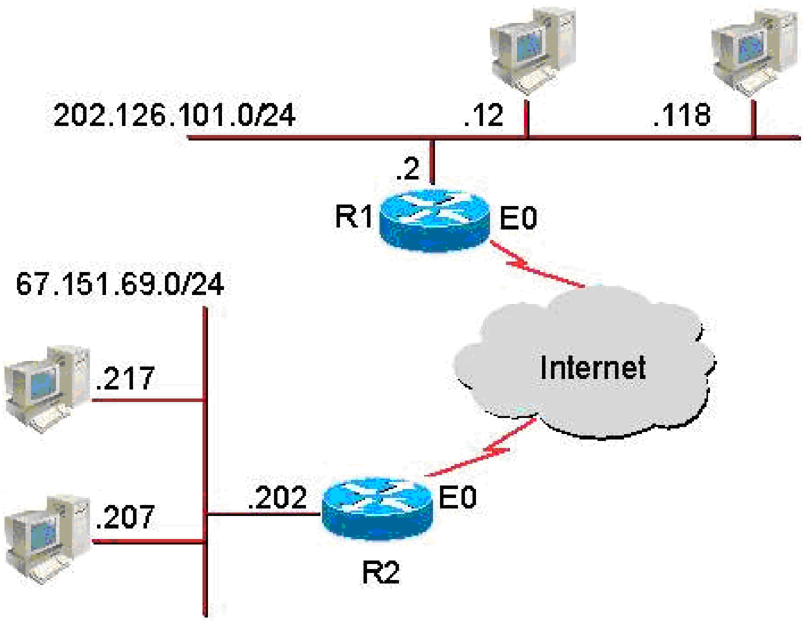
Caution The two ends of the tunnel must use the same key to communicate normally.

Caution Although every GRE packet includes this key when the encapsulation format is GRE, offering security on the basis of this key is not practicable.

Tunnel Interface Configuration Example

0 shows a network connection diagram.

Figure 4-2 Network Connection Diagram of the Tunnel Interface Configuration Example



In this configuration example, a tunnel is established between R1 and R2. The 202.126.101.0/24 subnet behind device R1 and the 67.151.69.0/24 subnet behind device R2 communicate with each other through the tunnel. The connection between R1 and R2 is transparent and invisible to external networks, that is, it is implemented through a VPN. The following example provides the related configuration of the tunnel between R1 and R2.

Related configuration on R1:

```
interface Tunnel0
ip address 21.21.21.3 255.255.255.0
tunnel source 179.208.12.221
tunnel destination 179.208.12.55
!
interface FastEthernet0/0
ip address 179.208.12.221 255.255.255.0
!
interface FastEthernet0/1
ip address 202.106.101.2 255.255.255.0
!
```

Related configuration on R2:

```
interface Tunnel0
ip address 21.21.21.5 255.255.255.0
tunnel source 179.208.12.55
tunnel destination 179.208.12.221
!
interface FastEthernet0/0
ip address 179.208.12.55 255.255.255.0
!
interface FastEthernet0/1
ip address 67.151.69.202 255.255.255.0
!
```

As shown in the above configuration, R1 and R2 both use the f0/0 Ethernet interface to establish a tunnel and use the f0/1 Ethernet interface to connect with external networks and serve as the gateway of the internal network.

Troubleshooting Tunnel Interfaces

If the two ends of a tunnel cannot communicate with each other normally, you can check the following aspects:

1. Ensure the two ends of the tunnel have connectable physical channels, that is, there is connectivity between them when no tunnel is used. In other words, the source address (local address) and the destination address (peer address) of the tunnel can normally communicate with each other.

2. Ensure that the source and destination addresses of the tunnel correspond with each other. In other words, the source address of the tunnel must match with the destination address of the tunnel.

3. Ensure that the tunnel has used the correct encapsulation format (defaulted to GRE) and the same encapsulation format is applied at both ends.

In addition, after the GRE is used as the encapsulation format of the tunnel, you must ensure that the two ends have the same check settings, key settings and reception rule settings.

Configuring Dialer Interfaces

Dialer Interface Overview

Ruijie products support physical dialer interfaces such as sync serial ports and async serial ports. The following section describes how to set the dialer interface. For other parameter settings, monitoring & maintenance, typical configuration example, and troubleshooting, see the dial-up configuration guide in this manual.

Configuring Dialer Interfaces

Ruijie products provide the **interface dialer** command to enter dialer interface configuration mode. Same as other logical interfaces, the first time you enter dialer interface configuration mode, you can create a dialer interface. Use the **no interface dialer** command to delete an existing dialer interface if necessary. For the parameter settings of the dialer interface, see the dial-up configuration section.

Command	Function
Ruijie(config)# **interface dialer** *dialer-number*	Enters dialer interface configuration mode.
Ruijie(config)# **no interface diale**r *dialer-number*	Deletes an existing dialer interface.

Monitoring and Maintaining Dialer Interfaces

Ruijie products provide the **show interfaces dialer** command for monitoring and maintaining dialer interfaces. If the link layer protocol is encapsulated on a dialer interface, use the **debug** command of the appropriate protocol to view the communication status of the dialer inteface.

Command	Function
R2501+# **show interfaces dialer** *dialer-number*	Shows the status of a dialer interface.

The following example shows how to use the **show interfaces dialer** command.

```
Ruijie# show interfaces dialer 1
dialer 1 is UP  (spoofing) , line protocol is UP  (spoofing)
Hardware is  dialer
Interface address is: 192.168.20.1/24
MTU 1500 bytes, BW 56 Kbit
Encapsulation protocol is SLIP, loopback not set
Keepalive interval is 0 sec , set
Carrier delay is 2 sec
RXload is 1 ,Txload is 1
Queueing strategy: WFQ
5 minutes input rate 0 bits/sec, 0 packets/sec
5 minutes output rate 0 bits/sec, 0 packets/sec
0 packets input, 0 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 abort
0 packets output, 0 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets
```

From the above information, you can know the settings and working status of the dialer interface, such as interface and link status, IP address settings, MTU settings, and bandwidth settings.

Configuring Sub-Interfaces

Sub-Interface Overview

A sub-interface is a logical interface deriving from an individual physical interface and depends on the physical interface. One physical interface may be configured with multiple sub-interfaces, providing great flexibility for application. A sub-interface is one of the logical interfaces derived from a physical interface. In other words, multiple logical interfaces are associated with a physical interface. The logical interfaces of a physical interface share the parameters of the physical interface, but they have their respective link layer and network layer configuration parameters.

The following physical interfaces support sub-interfaces:

- Fast Ethernet interfaces
- WAN interfaces encapsulated with Frame Relay
- WAN interfaces encapsulated with X.25

Sub-interface Configuration

Configuring the Sub-Interface of an Ethernet Interface

Ruijie products provide the **interface fastethernet** command to enter sub-interface configuration mode. Same as other logical interfaces, the first time you enter the sub-interface configuration mode, you can create a sub-interface. Use the **no interface fastethernet** command to delete an existing sub-interface if necessary. For the parameter settings of Ethernet sub-interfaces, see the sections about Ethernet interface configuration and IP protocol configuration.

Command	Function
Ruijie(config)# **interface fastethernet** *slot-number/interface-number.subinterface-number*	Enter s Ethernet sub-interface configuration mode.
Ruijie(config)# **no interface fastethernet** *slot-number /interface-number.subinterface-number*	Deletes an existing Ethernet sub-interface.

Where, the *slot-number/interface-number* is the sequence number of the slot or physical interface, and the *subinterface-number* is the sequence number of the sub-interface on the physical interface. Note that they are connected by a dot (**.**).

Configuring the Sub-Interface of a WAN Interface Encapsulated with Frame Relay

After a WAN interface is encapsulated with frame relay, Ruijie products provide the **interface serial** command for entering WAN sub-interface configuration mode. Same as other logical interfaces, the first time you enter the WAN sub-interface configuration mode, you can create a WAN sub-interface. Use the **no interface serial** command to delete an existing WAN sub-interface if necessary.

Command	Function
Ruijie(config)# **interface serial** *slot-number/interface-number. subinterface-number* [**point-to-point** | **multipoint**]	Enters WAN sub-interface configuration mode.
Ruijie(config)# **no interface serial** *slot-number/interface-number. subinterface-number* [**point-to-point** | **multipoint**]	Deletes an existing WAN sub-interface.

Where, the *slot-number/interface-number* is the sequence number of the slot or physical interface, and the *subinterface-number* is the sequence number of the sub-interface on the physical interface. Note that they are connected with a dot (**.**). In addition, the subinterface-number must be greater than 0.

After a sub-interface of the WAN encapsulated with frame relay is created, you can configure its IP address, virtual circuit, and frame relay address mapping. On the sub-interface, you can also configure the frame relay protocol and some other related parameters of its home physical interface. For the parameter settings of the WAN sub-interface, see the section about WAN protocol configuration.

Configuring the Sub-Interface of a WAN Interface Encapsulated with X.25

After a WAN interface is encapsulated with X.25, Ruijie products provide the **interface serial** command for entering WAN sub-interface configuration mode. Same as other logical interfaces, the first time you enter the WAN sub-interface configuration mode, you can create a WAN sub-interface. Use the **no interface serial** command to delete an existing WAN sub-interface if necessary.

Command	Function
Ruijie(config)# **interface serial slot-number/interface-number. subinterface-number** [**point-to-point** | **multipoint**]	Enters WAN sub-interface configuration mode.
Ruijie(config)# **no interface serial slot-number/interface-number. subinterface-number** [**point-to-point** | **multipoint**]	Deletes an existing WAN sub-interface.

Where, the *slot-number/interface-number* is the sequence number of the slot or physical interface, and the *subinterface-number* is the sequence number of the sub-interface on the physical interface. Note that they are connected with a dot (“**.**). In addition, the subinterface-number must be greater than 0.

After a sub-interface of the WAN encapsulated with X.25 is created, you can configure its IP address and X.25 address mapping. On the sub-interface, you can also configure the X.25 protocol and some other related parameters of its home physical interface. For the parameter settings of the WAN sub-interface, see the section about WAN protocol configuration.

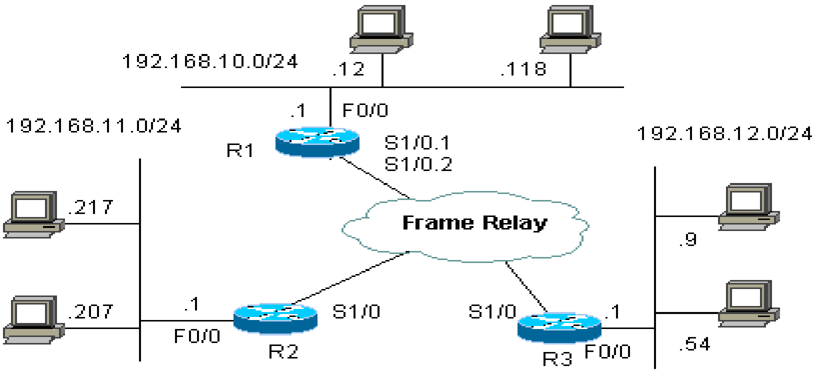
Monitoring and Maintaining Sub-Interfaces

Same as other interfaces, Ruijie products provide the show interfaces command to monitor and maintain sub-interfaces, but with the name of the home physical interface and the sequence number of the sub-interface. For the WAN sub-interfaces, you can use the debug command of the encapsulated protocol to monitor and maintain them. For the sub-interfaces of Ethernet interfaces, you can monitor and maintain them according to the procedure for monitoring and maintaining Ethernet interfaces.

Sub-interface Configuration Example

This configuration example involves a frame relay sub-interface, whose network connections are shown in 0.

Figure 4-3 Network Connections for a Frame Relay Sub-interface Configuration Example



In this configuration example, R1 establishes frame relay links with the S1/0 interface (IP: 101.92.67.2) of R2 and the S1/0 interface (IP: 101.92.68.2) of R3 through the frame relay sub-interfaces S1/0.1 (IP: 101.92.67.1, DLCI: 20) and S1/0.2 (IP: 101.92.68.1, DLCI: 30) respectively. R1, R2 and R3 respectively become the gateways for the network segments connected with their Ethernet interface f0/0. After appropriate routes are configured on devices R1, R2 and R3, the three sub-networks (192.168.10.0/24, 192.168.11.0/24 and 192.168.12.0/24) can communicate with one another. The following example only provides the related configuration of the frame relay subinterface on R1.

```
interface Serial1/0
no ip address
encapsulation frame-relay
frame-relay intf-type dce
!
interface Serial 1/0.1 point-to-point
ip address 101.92.67.1 255.255.255.0
frame-relay interface-dlci 20
!
interface Serial 1/0.2 point-to-point
ip address 101.92.68.1 255.255.255.0
frame-relay interface-dlci 30
            !
```

Troubleshooting Sub-Interfaces

For details about how to troubleshoot a sub-interface, see the section about WAN protocol configuration in this manual if the physical interface of the sub-interface is a WAN interface, or the section about IP protocol configuration in this manual if the physical interface of the sub-interface is an Ethernet interface.

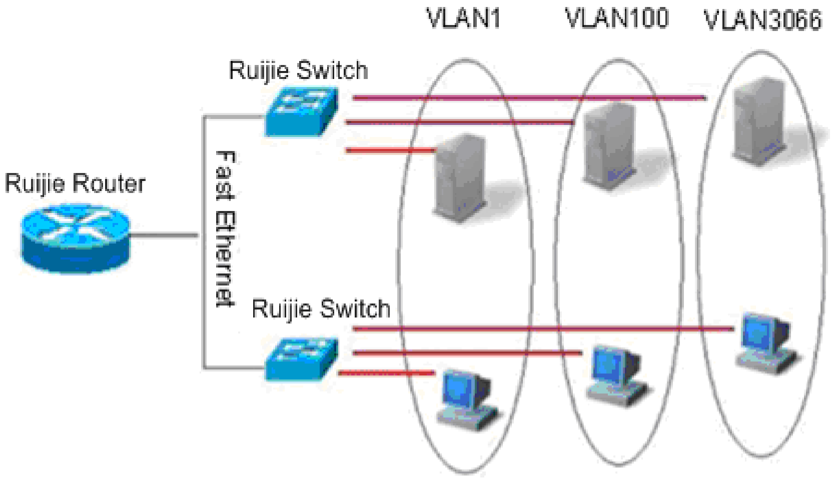
Configuring VLAN

This chapter describes how to configure IEEE802.1q VLAN.

Overview

Virtual Local Area Network (VLAN) is a logical network divided on a physical network. VLAN corresponds to the L2 network in the ISO model. The division of VLAN is not restricted by the physical locations of network ports. A VLAN has the same attributes as a common physical network. Except for no restriction on physical location, unicast, broadcast and multicast frames on layer 2 are forwarded and distributed within a VLAN, not being allowed to directly go to other VLANs.Therefore, when a host in a VLAN wants to communicate with another host in another VLAN, a layer 3 device must be used, as shown in the following diagram.

You can define a port as the member of a VLAN. All the terminals connected to the specified port are part of the VLAN. A network can support multiple VLANs. In this case, when you add, delete, and modify users in the VLANs, you do not need to modify the network configuration physically.



Like a physical network, a VLAN is usually connected to an IP subnet. A typical example is that all the hosts in the same IP subnet belong to the same VLAN. A layer 3 device must be used for communication between VLANs. Ruijie L3 devices can perform IP routing between VLANs through SVI (Switch Virtual Interfaces). For the configuration about SVI, refer to *Interface Management Configuration* and *IP Unicast Routing Configuration*.

Supported VLAN

Complying with IEEE802.1Q Standard, our products support up to 4094 VLANs(VLAN ID 1-4094), in which VLAN 1 is the default VLAN that cannot be deleted.

VLAN Member Type

You can determine the frames that can pass a port and the number of VLANs that the port can belong to by configuring the VLAN member type of the port. For the detailed description about VLAN member type , see the following table:

Member Type	Port Feature
Access	One access port can belong to only one VLAN, which must be specified manually.
Trunk (802.1Q)	By default, one Trunk port belongs to all the VLANs of the device itself, and it can forward the frames of all the VLANs. However, you can impose restriction by setting a list of allowed VLANs.

Configuring a VLAN

A VLAN is identified by its VLAN ID. You can add, remove, and modify the VLANs in the range of 2 to 4094 on a device. VLAN 1 is created by a device automatically and cannot be removed.

You can configure the member type of a port in a VLAN, add a port to a VLAN, and remove a port from a VLAN in the interface configuration mode.

Saving the VLAN Configuration

To save the VLAN configuration in the configuration file, execute the **copy running-config startup-config** command in the privileged EXEC mode. To view VLAN configuration, execute the **show vlan** command.

Default VLAN Configuration

The following table shows the default configuration of a VLAN.

Parameter	Default Value	Range
VLAN ID	1	1 to 4094
VLAN Name	VLAN xxxx, where xxxx is the VLAN ID	None
VLAN State	Active	Two status: active or inactive

Creating/Modifying a VLAN

In the privileged EXEC mode, you can create or modify a VLAN by executing the following commands.

Command	Function
Ruijie(config)# **vlan** *vlan-id*	Enter a VLAN ID. If you enter a new VLAN ID, the device will create it. If you enter an existing VLAN ID, the device modifies the corresponding VLAN.

Command	Function
Ruijie(config)# **name** *vlan-name*	(Optional) Name the VLAN. If you skip this step, the device automatically assigns the VLAN a name of VLAN xxxx, where xxxx is a 4-digit VLAN ID starting with 0. For example, VLAN 0004 is the default name of VLAN 4.

To restore the name of a VLAN to its default, simply enter the **no name** command.

The following example creates VLAN 888, names it test888, and saves its configuration into the configuration file:

```
Ruijie# configure terminal
Ruijie(config)# vlan 888
Ruijie(config-vlan)# name test888
Ruijie(config-vlan)# end
```

Deleting a VLAN

You cannot delete the default VLAN (VLAN 1).

In the privileged EXEC mode, you can delete a VLAN by executing the following command.

Command	Function
Ruijie(config)# **no vlan** *vlan-id*	Enter the VLAN ID that you want to delete.

Adding Existing Access Ports to Specified VLAN

If you assign a port to an inexistent VLAN, the switch will automatically create that VLAN.

In the privileged EXEC mode, you can assign a port to a VLAN by executing the following command.

Command	Function
Ruijie(config-if)# **switchport mode access**	Define the member type of the port in a VLAN (L2 ACCESS port).
Ruijie(config-if)# **switchport access vlan** *vlan-id*	Assign the port to the VLAN.

The following example adds Ethernet 1/10 to VLAN20 as an access port:

```
Ruijie# configure terminal
Ruijie(config)# interface fastethernet 1/10
Ruijie(config-if)# switchport mode access
Ruijie(config-if)# switchport access vlan 20
Ruijie(config-if)# end
```

The following example shows how to verify the configuration:

```
Ruijie(config)#show interfaces gigabitEthernet 3/1
 switchport
Switchport is enabled
Mode is access port
```

```
Acsess vlan is 1,Native vlan is 1
Protected is disabled
Vlan lists is ALL
```

Caution

In the R2700 switching card, although the access vlan can also be configured on the trunk port, the access port configuration does not take effect and the port remains in the trunk port. Only the configuration of native vlan and allowed vlan list takes effect.

Adding Access Ports to the Existing VLAN

In VLAN configuration mode, add the specified Access port to this VLAN. The effect of this command is the same as the command to specify the VLAN to which the interface belongs in interface configuration mode (namely **switchport access vlan** *vlan-id*).

Command	Function
Ruijie(config)# **vlan** *vlan-id*	Type in a VLAN ID. If a new VLAN ID is typed in, the device will create a VLAN. If an existing VLAN ID is typed in, the corresponding VLAN will be modified.
Ruijie(config-vlan)# **add interface** { *interface-id* | **range** *interface-range* }	Add one or a group of Access ports to the existing VLAN. By default, all layer-2 Ethernet ports belong to VLAN1.
Ruijie(config-vlan)# **[no]add interface** { *interface-id* | **range** *interface-range* }	Delete one or a group of Access ports form the existing VLAN.
Ruijie(config-vlan)#**show interface** *interface-id* **switchport**	Display the information about layer-2 interface.

Caution

This command only applies to Access port.

In terms of these two commands to add interface to the VLAN, the later configured command will override the previously configured command.

The following example adds Access port (GigabitEthernet 0/10) to VLAN20:

```
Ruijie# configure terminal
            SwitchA(config)#vlan 20
            SwitchA(config-vlan)#add interface GigabitEthernet 0/10
```

The following example how to verify the configurations:

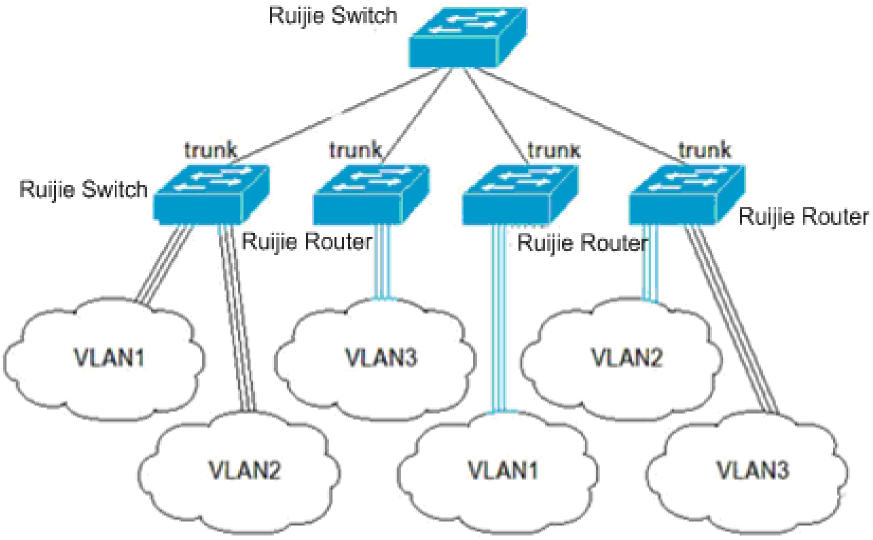
```
Ruijie# show interface GigabitEthernet 0/10 switchport
Interface        Switchport   Mode  Access Native Protected  VLAN lists
----------       --------     -----  ----- ------ ---------- -------
GigabitEthernet 0/10 enabled  ACCESS  20     1    Disabled   ALL
```

Configuring VLAN Trunks

Overview

A trunk is a point-to-point link that connects one or multiple Ethernet switching interfaces to other network devices (for instance, router). A trunk can transmit the traffics of multiple VLANs.

The Trunk encapsulation of Ruijie device is 802.1Q-complied. The following diagram shows a network connected with trunks.



You can set a common Ethernet port or aggregate port to be a trunk port. For the details of aggregate port, refer to *Configuring Aggregate Port*.

In order to switch an interface between the access mode and the trunk mode, use the **switchport mode** command:

Command	Function
Ruijie(config-if)# **switchport mode access**	Set an interface to the access mode
Ruijie(config-if)# **switchport mode trunk**	Set an interface to the Trunk mode

A native VLAN must be defined for a trunk port. The untagged packets received and sent through the port are deemed as the packets of the native VLAN. Obviously, the default VLAN ID of the port (that is, the PVID in the IEEE 802.1Q) is the VLAN ID of the native VLAN. Moreover, you must untag them before sending the packets of the native VLAN through the trunk port. The default native VLAN of a trunk port is VLAN 1.

When you configure a trunk link, be sure that the ports on both ends of the trunk belong to the same native VLAN.

Configuring a Trunk Port

Basic Trunk Port Configuration

In the privileged EXEC mode, you can configure a trunk port by executing the following command.

Command	Function
Ruijie(config-if)# **switchport mode trunk**	Configure the port as a L2 trunk port.
Ruijie(config-if)# **switchport trunk native vlan** *vlan-id*	Specify a native VLAN for the port.

To restore all the trunk-related settings of a trunk port to their defaults, use the **no switchport mode** command in the interface configuration mode.

Defining the Allowed VLAN List of a Trunk Port

By default, the traffic of all VLANs in the range of 1 to 4094 can be transmitted over a trunk port. However, you can restrict the traffic of some VLANs from passing the trunk port by setting its allowed VLAN list.

In the priviledged mode, you can modify the allowed VLAN list of a trunk port by executing the following command.

Command	Function
Ruijie(config-if)# **switchport trunk allowed vlan {all | [add | remove | except] }** *vlan-list*	(Optional) Configure the allowed VLAN list of the trunk port. The vlan-list parameter may be a VLAN or a series of VLANs. It starts with a small VLAN ID and ends with a large VLAN ID. Both IDs are connected with “-”, such as 10–20. All: Add all the allowed VLANs to the allowed VLAN list; add: Add the specified VLAN list to the allowed VLAN list; remove: Remove the specified VLAN list from the allowed VLAN list; except: Add all the VLANs other than the specified VLAN list to the allowed VLAN list.

To restore the allowed VLAN list of the trunk port to its default, execute the **no switchport trunk allowed vlan** command in the interface configuration mode.

The following example removes VLAN 2 from the allowed VLAN list of port 1/15:

```
Ruijie(config)# interface fastethernet 1/15
Ruijie(config-if)# switchport trunk allowed vlan remove 2
Ruijie(config-if)# end
Ruijie# show interfaces fastethernet 1/15 switchport
Interface Switchport Mode  Access Native Protected VLAN lists
--------- ---------- ----  ------ ------ -------- ----------
```

```
Gi0/15    enabled   TRUNK  1     1      Disabled   1,3－4094
```

Configuring a Native VLAN.

Tagged or untagged 802.1Q frames can be received or sent on a trunk port. Untagged frames are used to transmit the traffic of the native VLAN. By default, the native VLAN is VLAN 1.

In the privileged EXEC mode, you can configure a native VLAN for a trunk port by executing the following command.

Command	Function
Ruijie(config-if)# **switchport trunk native vlan** *vlan-id*	Configure a native VLAN.

To restore the native VLAN of a trunk port to its default, execute the **no switchport trunk native vlan** command in the interface configuration command.

If a frame carries the VLAN ID of the native VLAN, it will be automatically untagged when being forwarded through the trunk port.

When you set the native VLAN of a trunk port to an inexistent VLAN, the switch will not automatically create the VLAN. In addition, the native VLAN of a trunk port may be out the allowed VLAN list. In this case, the traffic of the native VLAN cannot pass the trunk port.

Showing VLAN Information

Only in the privileged EXEC mode can you view the VLAN information, including VLAN VID, VLAN status, member ports of the VLAN, and VLAN configuration. The related commands are listed as below:

Command	Function
show vlan [**id** *vlan-id*]	Show the information about all or the specified VLAN.

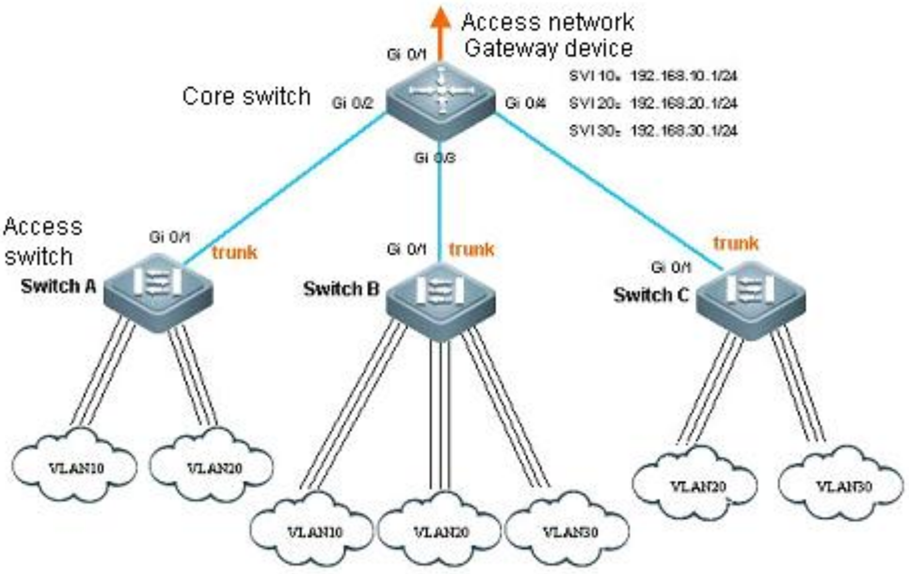
The following example shows the information about a VLAN:

```
Ruijie# show vlan
VLAN  Name      Status            Ports
                ---- --------  --------   ------------------------------
                  1 VLAN0001   STATIC     Gi0/1, Gi0/5, Gi0/6, Gi0/7
                                          Gi0/8, Gi0/9, Gi0/10, Gi0/11
                                          Gi0/12, Gi0/13, Gi0/14, Gi0/15
                                          Gi0/16, Gi0/17, Gi0/18, Gi0/19
                                          Gi0/20, Gi0/21, Gi0/22, Gi0/23
                                          Gi0/24
                 10 VLAN0010  STATIC    Gi0/2, Gi0/3
                 20 VLAN0020  STATIC    Gi0/2, Gi0/3, Gi0/4
                30 VLAN0030  STATIC   Gi0/3, Gi0/4

                Ruijie#show vlan id 20
                VLAN  Name              Status    Ports
                ----  --------         ------   ----------------------
```

```
20  VLAN0020          STATIC    Gi0/2, Gi0/3, Gi0/4
```

Configuration Examples

Network Topology



Networking Requirements

As shown above, an Intranet is divided into VLAN 10, VLAN 20 and VLAN 30 in order to realize laye-2 isolation. The IP subnets corresponding to three VLANs are 192.168.10.0/24, 192.168.20.0/24 and 192.168.30.0/24. The three VLANs are interconnected through the IP forwarding capacity of layer-3 core switch.

Configuration Tips

This example shows to how to configure the core switch and one of the access switches:

- Configure three VLANs on the core switch; configure the port connecting access switch to trunk port and specify the allowed vlan list to realize layer-2 isolation;
- Configure three SVI interfaces on the core switch to serve as the gateway interfaces for IP subnets corresponding to the three VLANs; configure the corresponding IP addresses;
- Create VLANs on three access switches and assign Access port for each VLAN; specify the trunk port for connecting core switch. This example shows the configuration steps on the access switch of Switch A.

Configuration Steps

Configurations on Core Switch

- Create VLAN

# Enter the global configuration mode

```
Ruijie#configure terminal
```

# Create VLAN 10

```
Ruijie(config)#vlan 10
```

# Create VLAN 20

```
Ruijie(config-vlan)#vlan 20
```

# Create VLAN 30

```
Ruijie(config-vlan)#vlan 30
```

# Return to the global configuration mode

```
Ruijie(config-vlan)#exit
```

- Configure respective trunk ports and specify the allowed vlan list

# Enter the interface range of Gi 0/2-4

```
Ruijie(config)#interface range GigabitEthernet 0/2-4
```

# Configure Gi 0/2-4 as trunk ports

```
Ruijie(config-if-range)#switchport mode trunk
```

# Return to the global configuration mode

```
Ruijie(config-if-range)#exit
```

# Enter port Gi 0/2

```
Ruijie(config)#interface GigabitEthernet 0/2
```

# Delete all vlans from the allowed vlan list of this port

```
Ruijie(config-if)#switchport trunk allowed vlan remove 1-4094
```

# Add vlan 10 and vlan 20 into the allowed vlan list of this port

```
Ruijie(config-if)#switchport trunk allowed vlan add 10,20
```

# Enter port Gi 0/3

```
Ruijie(config-if)#interface GigabitEthernet 0/3
```

# Delete all vlans from the allowed vlan list of this port

```
Ruijie(config-if)#switchport trunk allowed vlan remove 1-4094
```

# Add vlan 10, vlan 20 and vlan 30 into the allowed vlan list of this port

```
Ruijie(config-if)#switchport trunk allowed vlan add 10,20,30
```

# Enter port Gi 0/4

```
Ruijie(config-if)#interface GigabitEthernet 0/4
```

# Delete all vlans from the allowed vlan list of this port

```
Ruijie(config-if)#switchport trunk allowed vlan remove 1-4094
```

# Add vlan 20 and vlan 30 into the allowed vlan list of this port

```
Ruijie(config-if)#switchport trunk allowed vlan add 20,30
```

# Return to the global configuration mode

```
Ruijie(config-if)#exit
```

- Display vlan configurations on core switch

# Display vlan information, including vlan id, name, state and member ports

```
Ruijie#show vlan
VLAN  Name      Status             Ports
---- --------  --------   -------------------------------
   1 VLAN0001   STATIC    Gi0/1, Gi0/5, Gi0/6, Gi0/7
                          Gi0/8, Gi0/9, Gi0/10, Gi0/11
                          Gi0/12, Gi0/13, Gi0/14, Gi0/15
                          Gi0/16, Gi0/17, Gi0/18, Gi0/19
                          Gi0/20, Gi0/21, Gi0/22, Gi0/23
                          Gi0/24
```

```
 10 VLAN0010  STATIC     Gi0/2, Gi0/3
 20 VLAN0020  STATIC     Gi0/2, Gi0/3, Gi0/4
 30 VLAN0030  STATIC     Gi0/3, Gi0/4
```

# Display the vlan state of port Gi 0/2

```
Ruijie#show interface GigabitEthernet 0/2 switchport
Interface Switchport Mode  Access Native Protected VLAN lists
-------- ---------- -----  ------ ------ --------- ---------
Gi0/2     enabled   TRUNK   1      1     Disabled  10,20
```

# Display the vlan state of port Gi 0/3

```
Ruijie#show interface GigabitEthernet 0/3 switchport
Interface Switchport Mode  Access Native Protected VLAN lists
-------- ---------- -----  ------ ------ --------- ----------
Gi0/3     enabled   TRUNK   1      1     Disabled  10,20,30
```

# Display the vlan state of port Gi 0/4

```
Ruijie#show interface GigabitEthernet 0/4 switchport
Interface Switchport Mode  Access Native Protected VLAN lists
-------- ---------- -----  ------ ------ --------- ----------
Gi0/4     enabled   TRUNK   1      1     Disabled  20,30
```

- Create SVI port and specify the IP address

# Enter the global configuration mode

```
Ruijie#configure terminal
```

# Create SVI 10

```
Ruijie(config)#interface vlan 10
```

# Configure the IP address of SVI 10

```
Ruijie(config-if)#ip address 192.168.10.1 255.255.255.0
```

# Create SVI 20

```
Ruijie(config-if)#interface vlan 20
```

# Configure the IP address of SVI 20

```
Ruijie(config-if)#ip address 192.168.20.1 255.255.255.0
```

# Create SVI 30

```
Ruijie(config-if)#interface vlan 30
```

# Configure the IP address of SVI 30

```
Ruijie(config-if)#ip address 192.168.30.1 255.255.255.0
```

# Return to the global configuration mode

```
Ruijie(config-if)#exit
```

Configurations on the Access Switch of Switch A

- Create VLAN

# Enter the global configuration mode

- Ruijie#configure terminal

# Create VLAN 10

```
Ruijie(config)#vlan 10
```

# Create VLAN 20

```
Ruijie(config-vlan)#vlan 20
```

# Return to the global configuration mode

```
Ruijie(config-vlan)#exit
```

- Assign Access port for each VLAN

# Enter the interface range of Gi 0/2-12

```
Ruijie(config)#interface range GigabitEthernet 0/2-12
```

# Configure Gi 0/2-12 as Access ports

```
Ruijie(config-if)#switchport mode access
```

# Add Gi 0/2-12 to VLAN 10

```
Ruijie(config-if)#switchport access vlan 10
```

# Enter the interface range of Gi 0/13-24

```
Ruijie(config-if)#interface range GigabitEthernet 0/13-24
```

# Configure Gi 0/13-24 as Access ports

```
Ruijie(config-if)#switchport mode access
```

# Add Gi 0/13-24 to VLAN 20

```
Ruijie(config-if)#switchport access vlan 20
```

# Return to the global configuration mode

```
Ruijie(config-if)#exit
```

- Specify the trunk port for connecting core switch

# Enter port Gi 0/1

```
Ruijie(config)#interface GigabitEthernet 0/1
```

# Configure Gi 0/1 as trunk port

```
Ruijie(config-if)#switchport mode trunk
```

# Return to global configuration mode

```
Ruijie(config-if)#exit
```

Configuring RMON

Overview

RMON (Remote Monitoring) is a standard monitoring specification of IETF (Internet Engineering Task Force). It can be used to exchange the network monitoring data among various network monitors and console systems. In the RMON, detectors can be placed on the network nodes, and the NMS determines which information is reported by these detectors, for example, the monitored statistics and the time buckets for collecting history. The network device such as a router acts as a node on the network. The information of current node can be monitored by means of the RMON.

There are three stages in the development of RMON. The first stage is the remote monitoring of Ethernet. The second stage introduces the token ring which is referred to as the token ring remote monitoring module. The third stage is known as RMON2, which develops the RMON to a high level of protocol monitor.

The first stage of RMON (known as RMON1) contains nine groups. All of them are optional (not mandatory), but some groups should be supported by the other groups.

The switch implements the contents of Group 1, 2 , 3 and 9: the statistics, history, alarm and event.

Statistics

Statistics is the first group in RMON. It measures the basic statistics information of each monitored subnet. At present, only the Ethernet interfaces of network devices can be monitored and measured. This group contains a statistics of Ethernet, including the discarded packets, broadcast packets, CRC errors, size block, conflicts and etc.

History

History is the second group in RMON. It collects the network statistics information regularly and keeps them for processing later. This group contains two subgroups:

1. The subgroup History Control is used to set such control information as sampling interval and sampling data source.
2. The subgroup Ethernet History provides history data about the network section traffic, error messages, broadcast packets, utilization, number of collision and other statistics for the administrator.

Alarm

Alarm is the third group in RMON. It monitors a specific management information base (MIB) object at the specified interval. When the value of this MIB object is higher than the predefined upper limit or lower than the predefined lower limit, an alarm will be triggered. The alarm is handled as an event by means of recording the log or sending the SNMP Trap message.

Event

Event is the ninth group in RMON. It determines to generate a log entry or a SNMP Trap message when an event is generated due to alarms.

RMON Configuration Task List

Configuring Statistics

One of these commands can be used to add a statistic entry.

Command	Function
Ruijie(config-if)# **rmon collection stats** *index* [**owner** *ownername*]	Add a statistic entry.
Ruijie(config-if)# **no rmon collection stats** *index*	Remove a statistic entry.

⚠ **Caution**

The current version of Ruijie product supports only the statistics of Ethernet interface. The index value should be an integer between 1 to 65535. At present, at most 100 statistic entries can be configured at the same time.

Configuring History

One of these commands can be used to add a history entry.

Command	Function
Ruijie(config-if)# **rmon collection history** *index* [**owner** *ownername*] [**buckets** *bucket-number*] [**interval** *seconds*]	Add a history entry.
Ruijie(config-if)# **no rmon collection history** *index*	Remove a history entry.

⚠ **Caution**

The current version of Ruijie product supports only the records of Ethernet. The index value should be within 1 to 65535. At most 10 history entries can be configured.

Bucket-number: Specifies the used data source and time interval. Each sampling interval should be sampled once. The sampling results are saved. The Bucket-number specifies the maximum number of sampling. When the maximum is reached for the sampling records, the new one will overwrite the earliest one. The value range of Bucket-number is 1 to 65535. Its default value is 10.

Interval: Sampling interva in the range of 1 to 3600 seconds, 1800 seconds by default.

Configuring Alarm and Event

One of these command can be used to configure the alarm:

Command	Function

Command	Function
Ruijie(config)# **rmon alarm** *number variable interval* {**absolute** | **delta**} **rising-threshold** *value* [*event-number*] **falling-threshold** *value* [*event-number*] [**owner** *ownername*]	Add an alarm entry.
Ruijie(config)# **rmon event** *number* [**log**] [**trap** *community*] [**description** *description-string*] [**owner** *ownername*]	Add an event entry.
Ruijie(config)# **no rmon alarm** *number*	Remove an alarm.
Ruijie(config)# **no rmon event** *number*	Remove an event.

number: Alarm index in the range of 1 to 65535.
variable: Variable to be monitored by the alarm(in integer).
interval: Sampling interval in the range of 1 to 4294967295.
Absolute: each sampling value compared with the upper and lower limits.
Delta: the difference with previous sampling value compared with the upper and lower limits.
value: Upper and lower limits.
Event-number: when the value exceeds the upper or lower limit, the event with the index of Event-number will be triggered.
Log: Record the event.
Trap: Send the Trap message to the NMS when the event is triggered.
Community: Community string used for sending the SNMP Trap message.
Description-string: Description of the event.
Ownername: Owner of the alarm or the event.

Showing RMON status

Command	Function
Ruijie(config)# **show rmon alarms**	Show alarms.
Ruijie(config)# **show rmon events**	Show events.
Ruijie(config)# **show rmon history**	Show history.
Ruijie(config)# **show rmon statistics**	Show statistics.

RMON Configuration Examples

Example of Configuring Statistics

If you want to get the statistics of Ethernet Port 3 , use the following commands:

```
Ruijie(config)# interface gigabitEthernet 0/3
Ruijie(config-if)# rmon collection stats 1 owner aaa1
```

Example of Configuring History

Use the following commands if you want to get the statistics of Ethernet Port 3 every 10 minutes:

```
Ruijie(config)# interface gigabitEthernet 0/3
Ruijie(config-if)# rmon collection history 1 owner aaa1 interval 600
```

Example of Configuring Alarm and Event

If you want to configure the alarm function for a statistical MIB variable, the following example shows you how to set the alarm function to the instance ifInNUcastPkts.6 (number of non-unicast frames received on port 6; the ID of the instance is 1.3.6.1.2.1.2.2.1.12.6) in ***IfEntry*** table of MIB-II. The specific function is as follows: the switch checks the changes to the number of non-unicast frames received on port 6 every 30 seconds. If 20 or more than 20 non-unicast frames are added after last check (30 seconds earlier), or only 10 or less than 10 are added, the alarm will be triggered, and event 1 is triggered to do corresponding operations (record it into the log and send the Trap with “community” name as “rmon”). The “description” of the event is “ifInNUcastPkts is too much”). The “owner” of the alarm and the event entry is “aaa1”.

```
Ruijie(config)#rmon alarm 10 1.3.6.1.2.1.2.2.1.12.6 30 delta rising-threshold 20 1
falling-threshold 10 1 owner aaa1
Ruijie(config)#rmon event 1 log trap rmon description "ifInNUcastPkts is too much " owner aaa1
```

Example of Showing RMON Status

show rmon alarm

```
Ruijie# show rmon alarms
rmon alarm table:
                index: 10,
                interval: 30,
                oid = 1.3.6.1.2.1.2.2.1.12.6
                sampleType: 2,
                alarmValue: 0,
                startupAlarm: 3,
                risingThreshold: 20,
                fallingThreshold: 10,
                risingEventIndex: 1,
                fallingEventIndex: 1,
                owner: zhangesan,
                stats: 1,
```

show rmon event

```
Ruijie# show rmon events
rmon event table:
                index = 1
                description = ifInNUcastPkts
                type = 4
                community = rmon
                lastTimeSent = 0 d:0 h:0 m:0 s
                owner = zhangsan
                status = 1
```

show rmon history

```
Ruijie# show rmon history
rmon history control table:
                index = 1
                interface = FastEthernet 0/1
                bucketsRequested = 10
                bucketsGranted = 10
                interval = 1800
                owner = zhangsan
                stats = 1

rmon history table:
                index = 1
                sampleIndex = 198
                intervalStart = 0d:14h:0m:47s
                dropEvents = 0
                octets = 67988
                pkts = 726
                broadcastPkts = 502
                multiPkts = 189
                crcAllignErrors = 0
                underSizePkts = 0
                overSizePkts = 0
                fragments = 0
                jabbers = 0
                collisions = 0
                utilization = 0
```

show rmon statistics

```
Ruijie#  show rmon statistics
ether statistic table:
                index = 1
                interface = FastEthernet 0/1
                owner = zhangsan
                status = 0
                dropEvents = 0
                octets = 1884085
                pkts = 3096
                broadcastPkts = 161
                multiPkts = 97
                crcAllignErrors = 0
                underSizePkts = 0
                overSizePkts = 1200
                fragments = 0
                jabbers = 0
                collisions = 0
                packets64Octets = 128
                packets65To127Octets = 336
                packets128To255Octets = 229
                packets256To511Octets = 3
                packets512To1023Octets = 0
                packets1024To1518Octets = 1200
```

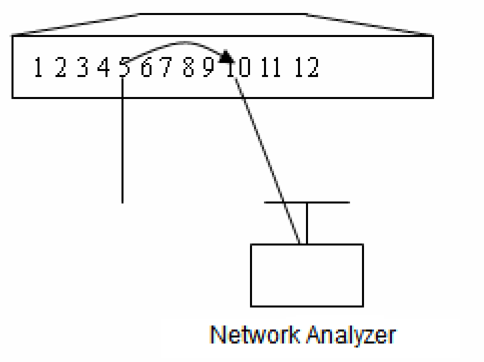
Configuring SPAN

Overview

With SPAN, you can analyze the communications between ports by copying a frame from one port to another port connected with a network analysis device or RMON analyzer. The SPAN mirrors all the packets sent/received at a port to a physical port for analysis.

For example, all the frames on the GigabitEhternet port 5 are mirrored to the GigabitEhternet port 10, as shown in the following figure. Although the network analyzer connected to port 10 is not directly connected to port 5, it can receive all the frames from port 5.

Figure 1-1 SPAN Configuration Example



SPAN does not affect the exchange of packets between the source and destination ports. Instead, it copies the frames incoming/outgoing the source port to the destination port. However, the frames may be discarded on an overflowed destination port, for example, when an 100Mbps port monitors an 1000Mbps port.

SPAN Concepts and Terms

SPAN Session

A SPAN session is a frame between a mirrored source port and destination port. It can monitor received, transmitted and bi-directional packets on one or multiple ports. Switched ports, Routed ports and AP ports can be configured as source and destination ports of SPAN sessions. Adding a port into a SPAN session will not affect the functioning of the switch.

You can configure a SPAN session on a disabled port, but the SPAN session is inactive. Only when the port is enabled, the SPAN session will become active. Moreover, the SPAN session will not take effect immediately after the switch is powered on. Only when the destination port becomes operable, the SPAN session will become active. You can use the **show monitor** [**session** *session_num*] command to view the status of a SPAN session.

Mirror Frame

Frame Direction

The SPAN session includes the following frame types:

- Received frames

Received frames include all known unicast frames and routing frames, and each received frame is copied to the destination port. In one SPAN session, you can monitor the the frames inputted from one or multiple source ports. Although a frame inputted from the source port is dropped due to some reasons, for example, port security, it is still sent to the destination port. This does not affect the function of the SPAN.

- Transmitted frames

All the frames sent from the source port are copied to the destination port. In one SPAN session, you can monitor the frames input from one or multiple source ports. If a frame from a port to the source port is dropped due to some reasons, the frame will not be sent to the destination port as well. Moreover, the format of the frames destined to the source port may change, for example, routed frames, source MAC address, destination MAC address, VLAN ID and TTL. Similarly, the format of the frames copied to the destination port will change.

- Bi-directional frames

Bi-directional frames include the above mentioned two frames. In one SPAN session, you can monitor the frames received and transmitted from/to one or multiple source ports.

SPAN Traffic

You can use the SPAN to monitor all network communications, including multicast frames and BPDU frames.

Source Port

The source port is also called the monitored port. In a SPAN session, frames on the source port are monitored for network analysis or troubleshooting. In a SPAN session, you can monitor received, transmitted and bi-directional frames and the maximum number of source ports is not limited.

A source port has the following features:

- It can be a switched port, routed port or AP.
- It cannot be a destination port at the same time.
- The source port and the destination port can reside in the same VLAN or different VLANs.

Destination Port

The SPAN session has a destination port (also known as the monitoring port) used to receive the frames copied from the source port.

The destination port has the following features:

- It can be a Switched Port , Routed Port or AP.
- The destination port can not be the source port at the same time.

Interaction between the SPAN and Other Functions

The SPAN interacts with the following functions.

- Spanning Tree Protocol (STP)
- The destination port of SPAN participates in the STP.

Configuring SPAN

Default SPAN Configuration

Function	Default Configuration
SPAN status	Disabled

Creating a SPAN Session and Specifying the Monitoring Port and Monitored Port

To set up a SPAN session and sepcify the destination port and the source port, execute the following commands.

Command	Function
Ruijie(config)# **monitor session** *session_number* **source interface** *interface-id* **[,| -] {both | rx | tx**}	Specify the source port. *interface-id*: Specify corresponding interface id.
Ruijie(config)# **monitor session** *session_num* **destination interface** *interface-id* {**encapsulation** | **switch**}	Specify the destination port. *interface-id*: Specify corresponding interface id. Add the encapsulation parameter to support the mirrored port encapsulation function, and add the switch parameter to support the mirrored destination port exchanging function.

To delete a SPAN session, use the **no monitor session** *session_number* command in the global configuration mode. To delete all the SPAN sessions, use the **no monitor session all** command in the global configuration mode. You can use the **no monitor session** *session_number* **source interface** *interface-id* command or the **no monitor session** *session_number* **destination interface** *interface-id* command to delete the source port or destination port in the global configuration mode.

The following example shows how to create session 1. First, clear the configuration of session 1, and then mirror the frames from port 1 to port 8. The **show monitor session** command allows you to verify your configuration.

```
Ruijie(config)# no monitor session 1
Ruijie(config)# monitor session 1 source interface gigabitEthernet 3/1 both
Ruijie(config)# monitor session 1 destination interface gigabitEthernet 3/8
Ruijie(config)# end
Ruijie# show monitor session 1
sess-num: 1
src-intf:
GigabitEthernet 3/1 frame-type Both

dest-intf:
```

```
GigabitEthernet 3/8
```

Caution Session 1 is used to support ther global cross-linecard port mirror.

Deleting a Port from the SPAN Session

To delete a port from a SPAN session, execute the following commands:

Command	Function
Ruijie(config)# **no monitor session** *session_number* **source interface** *interface-id* [,| **-] [both | rx | tx**]	Specify the source port to delete. *interface-id*: Specify corresponding interface id.

You can use the **no monitor session** *session_number* **source interface** *interface-id* command to delete the source port from a SPAN session in the global configuration mode. The following example shows how to delete gigabitEthernet 1/1 port from session 1 and check the configuration.

```
Ruijie(config)# no monitor session 1 source interface gigabitethernet 1/1 both
Ruijie(config)# end
Ruijie# show monitor session 1
sess-num: 1
dest-intf:
GigabitEthernet 3/8
```

Specifying the Source/Destination MAC for the Mirrored Frame

To specify the source and destination MAC addresses for the mirrored frames, execute the following commands:

Command	Function
Ruijie(config)# [**no**] **monitor session** *session_number* [**source interface** *interface-id* [**both** | **rx** | tx] | **destination interface** *interface-id*] | **mac** {**source** *mac-addr*| **destination** *mac-addr* } [**both** | **rx** | **tx**] {**source | destination**} *mac-address* [**both** | **rx** | **tx**]	Specify the source and destination MAC addresses for the mirrored frames.

You can use the **no monitor session** *session_number* **mac** {**source** \| **destionation**} [**both** \| **rx** \| **tx**] command to delete the source and destination MAC addresses for the mirrored frames.

Other Precautions

1) Connect the network analyzer to the monitoring port.
2) When the SPAN is enabled, the configuration change has the following result.

- If you change the VLAN configuration of the source port, the configuration takes effect immediately.
- If you change the VLAN configuration of the destination port, the configuration takes effect immediately.
- If you have disabled the source port or destination port, the SPAN does not take effect.

- If you add the source or destination port to an AP, this will remove the source port or destination port from the SPAN.

Displaying SPAN Configuration

The **show monitor** command shows the current SPAN status. The following example illustrates how to show the current status of SPAN session 1.

```
Ruijie# show monitor session 1
sess-num: 1
src-intf:
GigabitEthernet 3/1 frame-type Both
dest-intf:
GigabitEthernet 3/8
```

IP Address and Service Configuration

Configuring IP Address and Service

Understanding IP Address Configuration

Overview

An IP address is made up of 32 binary bits and expressed in dotted decimal format for the convenience of writing and description. In dotted decimal format, the 32 binary bits form four octets (1 octet equals to 8 bits). Each octet is separated by a period (dot) in the range from 0 to 255. For example, 192.168.1.1 is an IP address in dotted decimal format.

An IP address is an address that IP uses for interconnection. A 32-bit IP address consists of two parts: network address and local address. According to the first several bits of the network address of an IP address, IP addresses are divided into four categories.

Class A: There are totally 128 Class A IP addresses. The most significant bit is 0, followed by seven bits representing a network ID and 24 bits representing a host ID.

Figure 1-1

8 16 24 32

Class A IP address	0	Network ID	Host ID

Class B: There are totally 16,384 Class B IP addresses. The two most significant bits are 10, followed by 14 bits representing a network ID and 16 bits representing a host ID.

Figure 1-2

8 16 24 32

Class B IP address	1	0	Network ID	Host ID

Class C: There are totally 2,097,152 Class C IP addresses. The three most significant bits are 110, followed by 21 bits representing a network ID and 8 bits representing a host ID.

Figure 1-3

8 16 24 32

Class C IP address	1	1	0	Network ID	Host ID

Class D: The four most significant bits are 1110 and the other bits are a multicast IP address.

Figure 1-4

8 16 24 32

Class D IP address	1	1	1	0	Multicast address

Note

An IP address whose four most significant bits are 1111 is prohibited. This type of IP addresses, also called Class E IP addresses, is reserved.

When you build up a network, plan IP addresses according to the real network environment. To make the network connect to the Internet, you need to apply for IP addresses from a central authority, for example, China Internet Network Information Center (CNNIC) in China. Internet Corporation for Assigned Names and Numbers (ICANN) is the ultimate authority responsible for IP address allocation. However, for a private network, you do not need to apply for IP addresses. It is recommended that you assign private IP addresses for hosts in a private network.

The following table lists reserved and available addresses.

Class	Address Space	Status
Class A	0.0.0.0	Reserved
	1.0.0.0 to 126.0.0.0	Available
	127.0.0.0	Reserved
Class B	128.0.0.0 to 191.254.0.0	Available
	191.255.0.0	Reserved
Class C	192.0.0.0	Reserved
	192.0.1.0 to 223.255.254.0	Available
	223.255.255.0	Reserved
Class D	224.0.0.0 to 239.255.255.255	Multicast
Class E	240.0.0.0 to 255.255.255.254	Reserved
	255.255.255.255	Broadcast

Three blocks of IP addresses are reserved for private networks and are not used on the Internet. Address translation is required for a private network using one of these IP addresses to access the Internet. The following table lists these addresses, which are defined in RFC 1918.

Class	Address Space	Status
Class A	10.0.0.0 to 10.255.255.255	1
Class B	172.16.0.0 to 172.31.255.255	16
Class C	192.168.0.0 to 192.168.255.255	256

For information on the assignment of IP addresses, TCP/UDP ports and other codes, refer to RFC 1166.

IP Address Configuration Task List

The IP address configuration task list includes the following tasks. Only the first one is required, and the others are optional depending on your network requirements.

Assigning IP Addresses to Interfaces

Only hosts configured with IP addresses can receive and send IP packets. If an interface is configured with an IP address, this means that the interface supports the IP protocol.

Use the following commands to assign an IP address to an interface in interface configuration mode.

Command	Function
Ruijie(config-if)# **ip address** *ip-address mask*	Assigns an IP address to an interface.
Ruijie(config-if)# **no ip address**	Removes the IP address configured for the interface.

A 32-bit mask identifies the network part of an IP address. In a mask, the IP address bit corresponding to 1 represents the network ID and the IP address bit corresponding to 0 represents the host ID. For example, the mask corresponding to a Class A IP address is 255.0.0.0. You can subdivide a network into multiple segments by using the mask. The goal of subnet definition is to use some bits of the host address of an IP address as the network address to reduce the number of hosts and increase the number of networks. At this point, the mask is called a subnet mask.

Note

Theoretically, any bit of the host address of an IP address can be used as a subnet mask.

☑ Ruijie products only support continuous subnet masks from left to right starting from the network ID.

You can configure the following features related to IP addresses in a task list. These tasks are optional depending on your actual needs.

Assigning Multiple IP Addresses to an Interface

Ruijie products support assigning multiple IP addresses to an interface. Of the assigned IP addresses, one is the primary IP address and the others are secondary IP addresses. Theoretically, you can configure secondary addresses as many as you wish. A secondary IP address, however, must reside in different networks from the primary IP address or the other secondary IP addresses. The secondary IP address will be used frequently during the building of a network, for example, in the following cases:

- There are not enough host addresses for a network. For example, a LAN requires a Class C IP address to support up to 254 hosts. However, when there are more than 254 hosts in the LAN, another Class C IP address is necessary. Therefore, a host needs to connect two networks and thus needs configuring multiple IP addresses.
- Many older networks were built based on Layer 2 bridges without subnet definition. The use of secondary IP addresses makes it easy to upgrade such a network to an IP-based routing network. An IP address is assigned for every device in a subnet.
- Two subnets of a network might be separated by another network. By creating a subnet in each separated subnets, you can connect the two separated subnets together by assigning secondary IP addresses. One subnet cannot appear on two or more interfaces in a device.

Caution

Before configuring a secondary IP address, you need to confirm that the primary IP address has been configured. If a secondary IP address is configured on one device of the network, the secondary IP addresses configured for other devices of the network must be located in the same network. If an IP address is not assigned to other devices, you can configure the IP address as the primary IP address for a device.

Use the following command to assign a secondary IP address to an interface in interface configuration mode.

Command	Function
Ruijie(config-if)# **ip address** *ip-address mask* **secondary**	Assigns a secondary IP address to the interface.
Ruijie(config-if)# **no ip address** *ip-address mask* **secondary**	Removes the secondary IP address configured for the interface.

Configuring the Management IP Address and the Gateway Together

Ruijie Layer-2 switches allow you to configure the management IP address and the gateway by using the same command. Generally, Layer-2 switches provide the **ip default-gateway** command to configure a default gateway. Sometimes, the Layer-2 switch supports remote management via Telnet, and the management IP address and default gateway of the Layer-2 switch must be modified. In such a case, configuring either the **IP address** command or the **IP default-gateway** command will prevent you from configuring the other command (because the configuration has changed and this device can no longer be accessed via the network). Therefore, use the **gateway** keyword of the **IP address** command to modify the management IP address and default gateway.

Caution This command is only supported on Layer-2 devices.

Use the following commands to configure the management IP address and the gateway at the same time in interface configuration mode.

Command	Function
Ruijie(config-if)# **ip address** *ip-address mask* **gateway** *ip-address*	Configures the management IP address and gateway.
Ruijie(config-if)# **no ip address** *ip-address mask* **gateway**	Removes the configured management IP address and gateway.

Configuring the Address Resolution Protocol (ARP)

Every device in a LAN has two addresses: a local address and a network address. The local address is contained in the header of a frame at the data link layer. More exactly, it is a data link layer address. Since this local address is handled in the MAC sub-layer of the data link layer, it is normally called an MAC address representing an IP network device on a network. The network address represents a device on the Internet and indicates the network to which the device belongs.

For mutual communication, a device in a LAN must know the 48-bit MAC address of the other device. It can obtain the MAC address according to an IP address. This process is called ARP. The reversed ARP (RARP) can resolve the IP address based on a MAC address. You can resolve an address in two ways: ARP and proxy ARP. For more information on ARP, proxy ARP and RARP, refer to RFC 826, RFC 1027, and RFC 903.

ARP binds an IP to an MAC address. It can resolve the MAC address according to an IP address. Then, the relationship between the IP address and the MAC address is stored in the ARP cache. With the MAC address, a device can encapsulate the frames of the data link layer and send them to the LAN in the Ethernet II-type by default. However, the frames can also be encapsulated into other types of Ethernet frames, such as SNAP.

The principle of RARP is similar to ARP. RARP resolves the IP address according to an MAC address. RARP is applied on diskless workstations in general.

Normally, a device can work without any special address resolution configuration. Ruijie products can manage address resolution by the following configuration:

Configuring ARP Statically

ARP offers dynamic IP-to-MAC address mapping. It is not necessary to configure ARP statically in most cases. By configuring ARP Statically, Ruijie products can respond to ARP requests from other IP addresses.

Use the following commands to configure static ARP in global configuration mode.

Command	Function
Ruijie(config)# **arp** *ip-address mac-address arp-type*	Defines static ARP. Currently, Only *arp-type* can be set to **arpa** only.
Ruijie(config)# **no arp** *ip-address*	Removes static ARP.

Setting ARP Encapsulation

Currently Ruijie products only support Ethernet II ARP encapsulation, which also known as the **ARPA** keyword.

Setting ARP Timeout

ARP timeout takes effect for only the dynamically-learned IP-to-MAC address mapping. The shorter the timeout, the truer the mapping table saved in the ARP cache, but the more network bandwidth the ARP occupies. Hence the advantages and disadvantages should be weighted. Generally it is not necessary to configure the ARP timeout period unless otherwise stated.

Use the following commands to configure the ARP timeout period in interface configuration mode.

Command	Function
Ruijie(config-if)# **arp timeout** *seconds*	Configures the ARP timeout period in the range from 0 to 2147483, where 0 indicates no aging.
Ruijie(config-if)# **no arp timeout**	Removes the configuration.

By default, the timeout period is 3600 seconds.

Disabling IP Routing

IP routing is enabled by default. Do not execute this command unless you are sure that IP routing is not needed. Disabling IP routing will make the device lose all routes and the route forwarding function.

Run the following commands to disable IP routing in global configuration mode.

Command	Function
Ruijie(config)# **no ip routing**	Disables IP routing.

Command	Function
Ruijie(config)# **ip routing**	Enables IP routing

Configuring Broadcast Packet Processing

A broadcast packet is destined to all hosts in a physical network. Ruijie products support two kinds of broadcast packets: directed broadcast and flooding. A directed broadcast packet is sent to all the hosts in a specific network, with the host ID of the destination IP addresses set to all-1s. While a flooding broadcast packet is sent to all the hosts in all networks, with all 32 bits of the destination IP address set to 1s. Broadcast packets are currently misused by some protocols, including the Internet protocol. Therefore, it is a basic responsibility for a network administrator to manage and control broadcast packets.

Forwarding flooding broadcast packets may make the network overburden and thus affect network operation. This is known as broadcast storm. There are some ways to suppress and restrict broadcast storm in the local network. However, Layer 2 network devices like bridges and switches will forward and spread broadcast packets.

The best solution to solve broadcast storm is to specify a broadcast address for each network, that is, to implement directed broadcast. This requires the IP protocol to use directed broadcast instead of flooding broadcast as much as possible.

For details about broadcast, refer to RFC 919 and RFC 922.

You can configure the following features in a task list to process broadcast packets. These tasks are optional depending on actual network needs.

Enabling Directed Broadcast-to-Physical Broadcast Conversion

A directed broadcast IP packet is one destined to the broadcast address of an IP subnet. For instance, the packet destined to 172.16.16.255 is a directed broadcast packet. However, the node that generates this packet is not a member of the destination subnet.

Upon receipt of a directed broadcast IP packet, the device indirectly connecting the destination subnet will forward the packet in the same way as forwarding a unicast packet. After the directed broadcast IP packet arrives at a device directly connecting the subnet, the device transforms the packet into a flooding broadcast IP packet (whose destination address is all-1s in general), and then send it to all the hosts within the destination subnet by means of broadcast at the link layer.

Enabling the conversion from directed broadcast to physical broadcast on an interface allows the interface to forward directed broadcast IP packets to the directly connected network. This command only affects the transmission of the directed broadcast IP packets to the final destination subnet, but not other directed broadcast packets.

You can control the forwarding of directed broadcast IP packets as required on an interface by defining ACLs. Only those IP packets matching the ACLs will experience the conversion from directed broadcast to physical broadcast.

Use the following commands to configure directed broadcast-to-physical broadcast conversion in interface configuration mode.

Command	Function
Ruijie(config-if)# **ip directed-broadcast** [*access-list-number*]	Enables directed broadcast to physical broadcast conversion on the interface.

Command	Function
Ruijie(config-if)# **no ip directed-broadcast**	Disables the conversion.

Establishing an IP Broadcast Address

Currently, the most popular way is the destination address consisting of all 1s (255.255.255.255). Ruijie products can be configured to generate other IP broadcast addresses and receive all types of IP broadcast packets.

Use the following commands to set a broadcast IP address other than 255.255.255.255 in interface configuration mode.

Command	Function
Ruijie(config-if)# **ip broadcast-address** *ip-address*	Creates a broadcast address.
Ruijie(config-if)# **no ip broadcast-address**	Removes the configuration.

Monitoring and Maintaining IP Addresses

You can monitor and maintain networks according to the following task list. These tasks are optional depending on your actual needs.

Clearing Caches and Table Contents

You can remove all contents of a particular cache, table, or database, including

- Clearing the ARP cache
- Clearing the table of mapping between host names and IP addresses
- Clearing the routing table

Command	Function
Ruijie# **clear arp-cache**	Clears the ARP cache.
Ruijie# **clear ip route** {*network* [*mask*] | *}	Clears the IP routing table.

Displaying System and Network Status

You can show the contents of the IP routing table, cache, and database. Such information is very helpful in network troubleshooting. You also can display information about network reachability of a local device and determine the routing path that the packets of your device are taking after leaving the device.

Use the following commands to display system and network status information in privileged user mode.

Command	Function
Ruijie# **show arp** [[*ip* [*mask*] | *mac-address*] | **static** | **complete** | **incomplete**]	Shows the ARP cache table. The **complete** keyword is used to show resolved entries of dynamic ARP, and the **incomplete** keyword is used to show unresolved entries of dynamic ARP.
Ruijie# **show ip arp**	Shows the IP ARP cache table.
Ruijie# **show ip interface** [*interface-type interface-number*]	Shows interface IP address information.
Ruijie# **show ip route** [*network* [*mask*]]	Shows the routing table.
Ruijie#**show ip route**	Shows brief information about the routing table.

Command	Function
Ruijie# **ping** *ip-address* [**length** *bytes*] [**ntimes** *times*] [**timeout** *seconds*]	Tests network reachability.

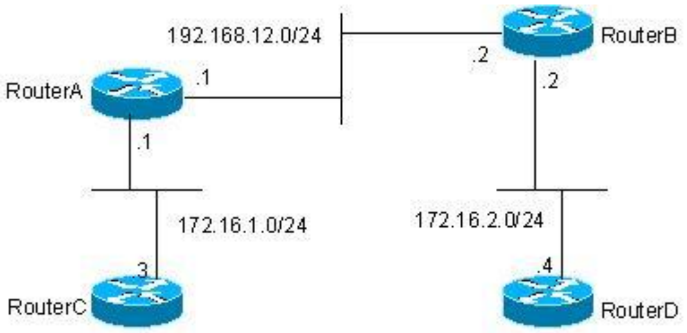
IP Address Configuration Examples

Example of Configuring Secondary IP Addresses

Configuration Requirements

The following figure shows IP address assignment and network device connections.

Figure 1-5 Examples of configuring secondary IP addresses.



Configure RIP. You can set the version to RIPv1 only. You can see the routes of 172.16.2.0/24 on router C and the routes of 172.16.1.0/24 on router D.

Configuration of the Routers

RIPv1 does not support classless routes. This means masks are not carried in route advertisements. 172.16.1.0/24 and 172.16.2.0/24 that belong to the same network are separated by the Class C network 192.168.12.0/24. Generally, router C and router D cannot learn detailed routes from each other. According to a feature of RIP, the mask of the route to be received should be set to the same value as that of the interface network if the route and the interface network belong to the same network. By configuring routers A and B, you can build a secondary network 172.16.3.0/24 on the network 192.168.12.0/24 to link the two separated subnets. The following presents a configuration description of routers A and B only.

Router A:

```
interface FastEthernet 0/0
ip address 172.16.3.1 255.255.255.0 secondary
ip address 192.168.12.1 255.255.255.0
!
interface FastEthernet 0/1
ip address 172.16.1.1 255.255.255.0
!
router rip
network 172.16.0.0
network 192.168.12.0
```

Router B:

```
interface FastEthernet 0/0
ip address 172.16.3.2 255.255.255.0 secondary
ip address 192.168.12.2 255.255.255.0
!
interface FastEthernet 0/1
ip address 172.16.2.1 255.255.255.0
!
router rip
network 172.16.0.0
network 192.168.12.0
```

Configuring the IP Service

IP Configuration Task List

IP service configuration covers the following configuration tasks. These tasks are optional depending on your actual needs.

Configuring the Default Gateway

☑ Run the command only on L2 devices.

If no destination IP address to which a packet will be sent is specified, the packet will be sent to the default gateway by default. Use the **show ip redirects** command to view the settings.

Use the following command to set the default gateway in global configuration mode. Use the **no** form of this command to remove the default gateway.

Command	Function
ip default-gateway *ip-address*	Sets the default gateway.

Use the following command to view the configured default gateway.

Command	Function
show ip redirects	Displays the default gateway.

Managing IP Connections

The IP protocol stack offers a number of services to control and manage IP connections. The Internet Control Message Protocol (ICMP) provides many of these services. Once a network problem occurs, a device or access server will send an ICMP message to the host or other devices. For detailed information on ICMP, see RFC 792.

Enabling the ICMP Destination Unreachable Message

When a router receives a non-broadcast packet destined to it and this packet uses an IP protocol that it cannot handle, it will return an ICMP destination unreachable message to the source address. Similarly, if the router is unable to forward the packet because it knows of no route to the destination address, it sends an ICMP host unreachable message. This feature is enabled by default.

Use the following command to enable or disable ICMP host unreachable messages in interface configuration mode.

Command	Function
Ruijie(config-if)# **ip unreachables**	Enables ICMP protocol unreachable and host unreachable messages.
Ruijie(config-if)# **no ip unreachables**	Disables ICMP protocol unreachable and host unreachable messages.

Enabling the ICMP Redirect Message

Routes are sometimes less than optimal. For example, it is possible for the device to be forced to resend a packet through the same interface on which the packet was received. If the device resends a packet through the receiving interface, it sends an ICMP redirect message to the originator of the packet, telling the originator that the gateway to this destination address is another device in the same subnet. Therefore, the originator will transmit subsequent packets based on the optimized path. This feature is enabled by default.

Use the following commands to enable or disable the ICMP redirect message in interface configuration mode.

Command	Function
Ruijie(config-if)# **ip redirects**	Enables the ICMP redirect message. It is enabled by default.
Ruijie(config-if)# **no ip redirects**	Disables the ICMP redirect message.

Enabling the ICMP Mask Reply Message

Occasionally, a network device needs to know the mask of a subnet on the Internet. To obtain this information, the device can send an ICMP mask request message. The device receiving the request will return an ICMP mask reply message. Ruijie products can respond to the ICMP mask request message. This function is enabled by default.

Use the following commands to enable or disable the ICMP mask reply message in interface configuration mode.

Command	Function
Ruijie(config-if)# **ip mask-reply**	Enables the ICMP mask reply message.
Ruijie(config-if)# **no ip mask-reply**	Disables the ICMP mask reply message.

Setting the IP MTU

All interfaces have a default MTU value. All packets which are larger than the MTU have to be fragmented before they are sent on an interface. Otherwise the packets cannot be forwarded on the interface.

Ruijie products allow you to adjust the MTU on an interface. Changing the MTU value can affect the IP MTU value, and the IP MTU value will be modified automatically to match the new MTU. However, changing the IP MTU value has no influence on the MTU value of the interface.

The interfaces of a device in a physical network should have the same MTU for the same protocol.

Use the following commands to set the IP MTU in interface configuration mode.

Command	Function
Ruijie(config-if)# **ip mtu** *bytes*	Sets the MTU in the range from 68 to 1500 bytes.
Ruijie(config-if)# **no ip mtu**	Restores the default MTU settings.

Configuring IP Source Routing

Ruijie products support IP source routing. Upon receiving an IP packet, the device will check its IP header like strict source route, loose source route and recorded route, which are defined in RFC 791. If one of these options is enabled, the device performs an appropriate action. Otherwise, it sends an ICMP error message to the source and then discards the packet. Ruijie products support IP source routing by default.

Use the following commands to enable or disable IP source routing in interface configuration mode.

Command	**Function**
Ruijie(config)# **ip source-route**	Enables IP source routing.
Ruijie(config)# **no ip source-route**	Disables IP source routing.

Configuring VRF

VRF Overview

Virtual Private Networks (VPNs) provide a secure way to share bandwidths in the backbone networks of ISPs. One VPN is the collection of the sites sharing routes. The sites connect to the service provider’s network through one or multiple interface links, with one VPN routing table associated with one interface. The VPN routing table is also referred to as a VPN routing or forwarding (VRF) table.

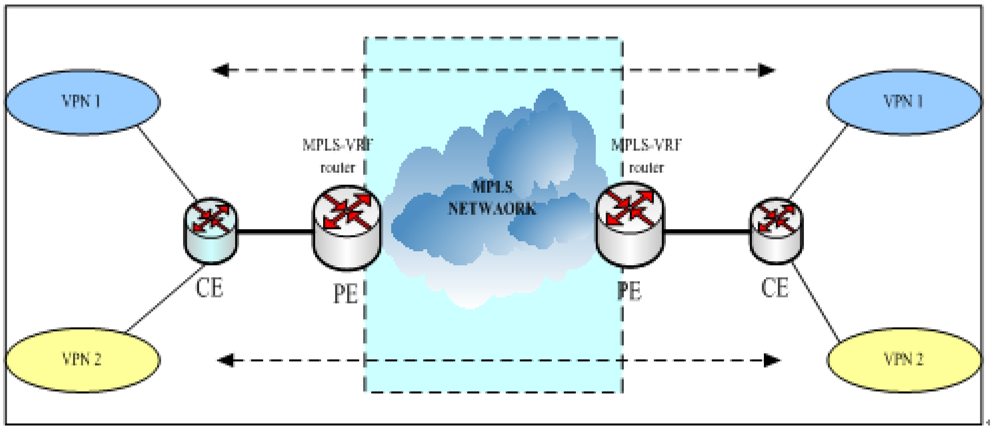
Caution

Ruijie RSRs support the VRF-lite feature, which is also known as the multi-VRF CE or multi-VRF Customer Edge Device, and can implement multiple VPN route forwarding instances when serving as CEs.

Working Principles of VRF-Lite

The working principles of VRF-lite are as follows:

- CEs provide multiple access channels to PEs to support user access. CEs advertise local routes to PEs, and learns VPN remote routes from PEs.
- PEs exchange routing information with CEs via static routing and dynamic routing protocols, such as BGP, RIP, and OSPF.
- PEs may have multiple interfaces that belong to one VPN. PEs exchange VPN routing information with each other via BGP.
- PEs do not rely on the functions of CEs.
- P devices do not process VPN information. That is, VPN information is transparent to P devices.



Typical Application Model of VRF-Lite

The packet processing flow is as follows when VRF-lite is enabled on a network:

- When a CE receives a packet from a VPN, it checks interface information to query the related VRF table. If a route is successfully found in the VRF table, the CE sends the packet to a PE.
- When the ingress PE receives a packet from a CE, it queries the VRF table. If a route is successfully found in the VRF table, the ingress PE adds an MPLS label to the route and sends the packet to the MPLS network.
- When the egress PE receives an MPLS packet from the MPLS network, it removes the MPLS label and finds the related VPN routing table. The egress PE searches for a common route. If a route is successfully found, the egress PE sends the packet to the related adjacency.
- When a CE receives a packet from the egress PE, it checks the inbound interface of the packet to obtain the related VPN routing table and then searches for a route. If a route is successfully found, the packet is sent to the VPN.

Multi-Protocol VRF

You can create only single-protocol VRF supporting IPv4 in version earlier than 10.4(3). Use the **ip vrf** command to create a single-protocol IPv4 VRF. To enable single-protocol IPv4 VRF on a network interface, use the **ip vrf forwarding** command in interface configuration mode.

To support IPv6 VPNs, the 10.4(3) version has introduced multi-address-family VRFs. A multi-address-family VRF enables you to define multiple address families under the same VRF. You can apply a multi-address-family VRF to both IPv4 VPNs and IPv6 VPNs. A multi-address-family VRF is also called a multi-protocol VRF. You can use the **vrf definition** command to create a multi-protocol VRF. To enable a multi-protocol VRF on a network interface, use the **vrf forwarding** command in interface configuration mode.

It is not allowed to use the commands for configuring a single-protocol IPv4 VRF to configure a multi-protocol VRF, and vice versa.

For example, you can use the configuration command of a multi-protocol VRF to create a multi-protocol VRF named "vrf1".

```
Ruijie(config)#vrf definition vrf1
```

If you try to use a configuration command of a single-protocol VRF to edit vrf1, the following prompting message will be displayed:

```
Ruijie(config)#ip vrf vrf1
% Use ‘vrf definition vrf1’ command.
```

If you try to use a multi-protocol VRF command to edit the single-protocol VRF named "vrf2" which only supports IPv4, the following prompting message will be displayed:

```
Ruijie(config)# vrf definition vrf2
% Use ‘ip vrf vrf2’ command.
```

The configuration commands (ip vrf and ip vrf forwarding) of single-protocol IPv4 VRFs will be reserved for a period of time until they are abandoned.

Configuring VRF-Lite

You can configure VRF-Lite as follows:

- A CE supports multiple users via VRF-Lite. Each user has its own routing table on the CE.
- Since each user has its own routing table, they may use the same IP address. This function is temporarily not supported.
- Multiple users share the physical line between a CE and a PE, and there are multiple logical interfaces on the physical line. The physical line can be implemented by multiple means.
- VRF-lite does not support functions related to MPLS-VRF. It mostly acts on CEs.
- For a PE, connecting it to multiple CEs does not differ from using VRF-lite on it.
- EBGP is recommended for route exchange between a PE and a CE. Of course, OSPF, RIP, and static routing protocols may also be used for route exchange, but that would be more complex. If the OSPF protocol is used for route exchange, you need to exercise caution during configuration. It is recommended that you enable the capability VRF-lite function when using the OSPF protocol for route exchange.

Creating VRF

Use the following commands to create a single-protocol IPv4 VRF.

Command	Function
Ruijie(config)# **ip vrf** *vrf-name*	Creates a VRF. vrf-name shall not exceed 31 characters.
Ruijie(config)# **no ip vrf** *vrf-name*	Deletes the VRF.

Use the following commands to create a multiprotocol VRF.

Command	Function
Ruijie(config)# **vrf definition** *vrf-name*	Creates a multi-protocol VRF. *vrf-name* shall not exceed 31 characters.
Ruijie(config-vrf)# **address-family ipv4**	Configures an IPv4 address family, namely to enable the IPv4 protocol of VRFs. No IPv4 address family is configured by default.
Ruijie(config-vrf-af)# **exit-address-family**	Exits VRF address-family configuration sub-mode.
Ruijie(config-vrf)# **address-family ipv6**	Configures an IPv6 address family, namely to enable the IPv6 protocol of VRFs. No IPv6 address family is configured by default.
Ruijie(config-vrf-af)# **exit-address-family**	Exits VRF address-family configuration sub-mode.

Configuring a VRF Descriptor

Command	Function
Ruijie(config-vrf)# **description** *string*	Configures a VRF descriptor, which contains at most 244 characters.

The following example configures a VRF descriptor named “vpn-a” for a single-protocol IPv4 VRF named “vrf1”.

```
Ruijie(config)#ip vrf vrf1
Ruijie(config-vrf)#description vpn-a
```

The following example configures a VRF descriptor named “vpn-b” for a multiprotocol VRF named “vrf2”.

```
Ruijie(config)#vrf definition vrf2
Ruijie(config-vrf)#description vpn-b
```

Enabling VRF on Interfaces

Command	Function
Ruijie(config-if)# **ip vrf forwarding** *vrf-name*	Binds an interface to a single-protocol IPv4 VRF. Note that *vrf-name* cannot be a multiprotocol VRF. If the IPv6 function does not need to be enabled on an interface, you can bind the interface to a single-protocol IPv4 VRF. If the IPv6 function needs to be enabled on an interface, it is not recommended that you use this command to bind the interface to a single-protocol IPv4 VRF.
Ruijie(config-if)# **no ip vrf forwarding** *vrf-name*	Cancels the binding between an interface and a single-protocol IPv4 VRF.
Ruijie(config-if)# **vrf forwarding** *vrf-name*	Binds an interface to a multiprotocol VRF. Note that *vrf-name* cannot be a single-protocol IPv4 VRF. If the IPv6 function needs to be enabled on an interface, it is recommended that you bind the interface to a multiprotocol VRF instead of a single-protocol IPv4 VRF.
Ruijie(config-if)# **no vrf forwarding** *vrf-name*	Cancels the binding between an interface and a multiprotocol VRF.

An interface does not belong to any VRF by default. That is, global routing applies.

Caution

After you bind an interface to a single-protocol IPv4 VRF, the IPv4 address originally configured for the interface will be invalid. The binding does not affect the IPv6 address configured for the interface. In general, enable VRF first on the interface and then configure an IPv4 address.

Caution

If you bind an interface to a single-protocol IPv4 VRF and enables the IPv6 protocol on the interface, the switch cannot forward IPv6 packets received on the interface. Therefore, it is recommended that you use the multiprotocol VRF if you want to bind the interface to a VRF and enable the IPv6 protocol on the interface at the same time.

Caution

You cannot bind an interface to a multiprotocol VRF not configured with any address family.

Caution

If you bind an interface to a multiprotocol VRF, all existing IPv4, IPv6, VRRP IPv4, and VRRP IP6 addresses configured for the interface will be deleted. In addition, the IPv6 protocol will be disabled on the interface.

Caution

If you bind an interface to a multiprotocol VRF not configured with any IPv4 address family, you cannot configure any IPv4 or VRRP IPv4 address for the interface. Before configuring an IPv4 or VRRP IPv4 address for the interface, you needs to configure an IPv4 address family for the multiprotocol VRF.

Caution

If you bind an interface to a multiprotocol VRF not configured with any IPv6 address family, you cannot configure any IPv6 or VRRP IPv6 address for the interface. Before configuring an IPv6 or VRRP IPv6 address for the interface, you needs to configure an IPv6 address family for the multiprotocol VRF.

Caution

If you delete the IPv4 address family configured for a multiprotocol VRF, all IPv4 and VRRP IPv4 addresses of all network interfaces bound to this VRF will be deleted, so will the static IPv4 routes whose routing VRF or next-hop VRF is this VRF. If you delete the IPv6 address family configured for a multiprotocol VRF, all IPv6 and VRRP IPv6 addresses of all network interfaces bound to this VRF will be deleted, the IPv6 protocol will be disabled on the interfaces, and all the static IPv6 routes whose routing VRF or next-hop VRF is this VRF will be deleted.

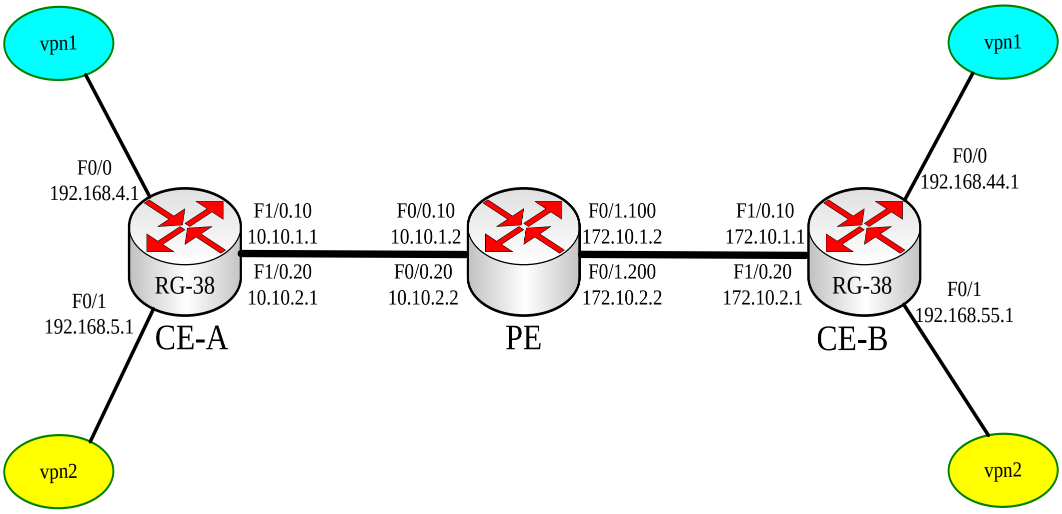
Configuring VRF Routes

Command	Function
Ruijie(config)# **ip route vrf** *vrf-name network mask interface nexthop*	Adds a route.
Ruijie(config)# **no ip route vrf** *vrf-name network mask*	Deletes a route.

You can also use routing protocols to configure routes.

VRF-Lite Configuration Examples

Example 1: As shown in the following figure, Ruijie devices serve as CEs and PEs. The CEs access two VPNs named “vpn1” and “vpn2”.



```
Ruijie# hostname CE-A
# Name the router.
CE-A# configure terminal
# Enter global configuration mode.
CE-A(config)# ip vrf vpn1
# Create a VRF named “vpn1”.
CE-A(config)# ip vrf vpn2
# Create a VRF named “vpn2”.
CE-A(config)# interface f0/0
# Enter interface configuration mode.
CE-A(config-if)#description  connecting-to-vpn1
# Link to vpn1.
CE-A(config-if)# ip vrf forwarding vpn1
# Enable VRF on the interface.
CE-A(config-if)# ip address 192.168.4.1 255.255.255.0
# Configure an IP address.
CE-A(config)# interface f0/1
# Enter interface configuration mode.
CE-A(config-if)# ip vrf forwarding vpn2
# Enable VRF on the interface.
CE-A(config-if)# ip address 192.168.5.1 255.255.255.0
# Configure an IP address.
CE-A(config-if)#description  connecting-to-vpn2
# Link to vpn2.
CE-A(config)# interface f1/0
# Enter interface configuration mode.
CE-A(config-if)#  no ip address
CE-A(config)# interface f1/0.10
# Enter a subinterface.
CE-A(config-if)# encapsulation dot1Q 10
# Encapsulate 802.1Q in VLAN 10.
```

```
CE-A(config-if)# ip vrf forwarding vpn1
# Enable VRF on the interface.
CE-A(config-if)# ip address 10.10.1.1 255.255.255.0
# Configure the IP address of the subinterface.
CE-A(config)# interface f1/0.20
# Enter a subinterface.
CE-A(config-if)# encapsulation dot1Q 20
# Encapsulate 802.1Q in VLAN 20.
CE-A(config-if)# ip vrf forwarding vpn2
# Enable VRF on the interface.
CE-A(config-if)# ip address 10.10.2.1 255.255.255.0
# Configure the IP address of the subinterface.
CE-A(config)# ip route vrf vpn1 192.168.44.0 255.255.255.0 10.10.1.2
# Configure a static route of vpn1.
CE-A(config)# ip route vrf vpn1 192.168.55.0 255.255.255.0 10.10.2.2
# Configure a static route of vpn2.
Ruijie# hostname CE-B
# Name the router.
CE-B# configure terminal
# Enter global configuration mode.
CE-B(config)# ip vrf vpn1
# Create a VRF named “vpn1”.
CE-B(config)# ip vrf vpn2
# Create a VRF named “vpn2”.
CE-B(config)# interface f0/0
# Enter interface configuration mode.
CE-Bconfig-if)# ip vrf forwarding vpn1
# Enable VRF on the interface.
CE-B(config-if)# ip address 192.168.44.1 255.255.255.0
# Configure an IP address.
CE-B(config-if)# description  connecting-to-vpn1
# Link to vpn1.
CE-B(config)# interface f0/1
# Enter interface configuration mode.
CE-B(config-if)# ip vrf forwarding vpn2
# Enable VRF on the interface.
CE-B(config-if)# ip address 192.168.55.1 255.255.255.0
# Configure an IP address.
CE-Bconfig-if)# description  connecting-to-vpn2
# Link to vpn2.
CE-B(config)# interface f1/0
# Enter interface configuration mode.
CE-B(config-if)#  no ip address
CE-B(config)# interface f1/0.10
# Enter a subinterface.
```

```
CE-B(config-if)# encapsulation dot1Q 100
# Encapsulate 802.1Q in VLAN 100.
CE-B(config-if)# ip vrf forwarding vpn1
# Enable VRF on the interface.
CE-B(config-if)# ip address 172.10.1.1 255.255.255.0
# Configure the IP address of the subinterface.
CE-B(config)# interface f1/0.20
# Enter a subinterface.
CE-B(config-if)# encapsulation dot1Q 200
# Encapsulate 802.1Q in VLAN 200.
CE-B(config-if)# ip vrf forwarding vpn2
# Enable VRF on the interface.
CE-B(config-if)# ip address 172.10.2.1 255.255.255.0
# Configure the IP address of the subinterface.
CE-B(config)# ip route vrf vpn1 192.168.4.0 255.255.255.0 172.10.1.2
# Configure a static route of vpn1.
CE-B(config)# ip route vrf vpn1 192.168.5.0 255.255.255.0 172.10.2.2
# Configure a static route of vpn2.
# Next, configure PEs.
Router# configure terminal
Router(config)# ip vrf v1
Router(config-vrf)# rd 100:1
Router(config-vrf)# route-target export 100:1
Router(config-vrf)# route-target import 100:1
Router(config-vrf)# exit
Router(config)# ip vrf v2
Router(config-vrf)# rd 100:2
Router(config-vrf)# route-target export 100:2
Router(config-vrf)# route-target import 100:2
Router(config-vrf)# exit
Router(config)# ip cef
Router(config)# interface FastEthernet0/0.10
Router(config-if)# encapsulation dot1q 10
Router(config-if)# ip vrf forwarding v1
Router(config-if)# ip address 10.10.1.2 255.255.255.0
Router(config-if)# exit
Router(config)# interface FastEthernet0/1.100
Router(config-if)# encapsulation dot1q 100
Router(config-if)# ip vrf forwarding v1
Router(config-if)# ip address 172.10.1.2 255.255.255.0
Router(config-if)# exit
Router(config)# interface FastEthernet0/0.20
Router(config-if)# encapsulation dot1q 20
Router(config-if)# ip vrf forwarding v2
Router(config-if)# ip address 10.10.2.2 255.255.255.0
Router(config-if)# exit
Router(config)# interface FastEthernet0/1.200
Router(config-if)# encapsulation dot1q 200
Router(config-if)# ip vrf forwarding v2
Router(config-if)# ip address 172.10.2.2 255.255.255.0
Router(config-if)# exit
Router(config)# ip route vrf v1 192.168.4.0 255.255.255.0 10.10.1.1
Router(config)# ip route vrf v1 192.168.44.0 255.255.255.0 172.10.1.1
Router(config)# ip route vrf v2 192.168.5.0 255.255.255.0 10.10.2.1
Router(config)# ip route vrf v2 192.168.55.0 255.255.255.0 172.10.2.1


Example 2 : Configure a simple multiprotocol VRF.
# Create a multiprotocol VRF.
Ruijie(config)#vrf definition multi-af-vrf-example
# Configure a VRF descriptor.
Ruijie(config-vrf)#description This-is-an-example
# Configure an IPv4 address family.
Ruijie(config-vrf)#address-family ipv4
Ruijie(config-vrf-af)#exit-address-family
# Configure an IPv6 address family.
Ruijie(config-vrf)#address-family ipv6
Ruijie(config-vrf-af)#exit-address-family
Ruijie(config-vrf)#interface VLAN 1
# Bind VLAN 1 to a multiprotocol VRF.
Ruijie(config-if-VLAN 1)#vrf forwarding multi-af-vrf-example
Ruijie(config-if-VLAN 1)#ip address 1.1.1.1 255.255.255.0
Ruijie(config-if-VLAN 1)#ipv6 address 1000::1/64
Ruijie(config-if-VLAN 1)#exit
# Configure a static IPv4 or IPv6 route of the multiprotocol VRF.
Ruijie(config)#ip route vrf multi-af-vrf-example 0.0.0.0 0.0.0.0 1.1.1.2
Ruijie(config)#ipv6 route vrf multi-af-vrf-example ::/0 1000::2
```

VRF-Lite Debugging

Use the following command to check the routing table of a VRF.

Command	Function
show ip route vrf *vrf-name*	Displays the routes of the specified VRF.

For details about the command syntax, see the *VRF Command Reference*.

Use the following command to clear the routing table of a VRF.

Command	Function

clear ip route vrf *vrf-name* *	Clears the routes of the specified VRF.

For details about the command syntax, see the *VRF Command Reference*.

Use the following commands to check information about a VRF in the system.

Command	Function
show ip vrf [*vrf-name*]	Displays information about an IPv4 VRF.
show vrf [brief] [vrf-name]	Displays brief information about a VRF.
show vrf ipv4 [*vrf-name*]	Displays brief information about an IPv4 VRF.
show vrf ipv6 [*vrf-name*]	Displays brief information about an IPv6 VRF.
show vrf detail [*vrf-name*]	Displays details about a VRF.

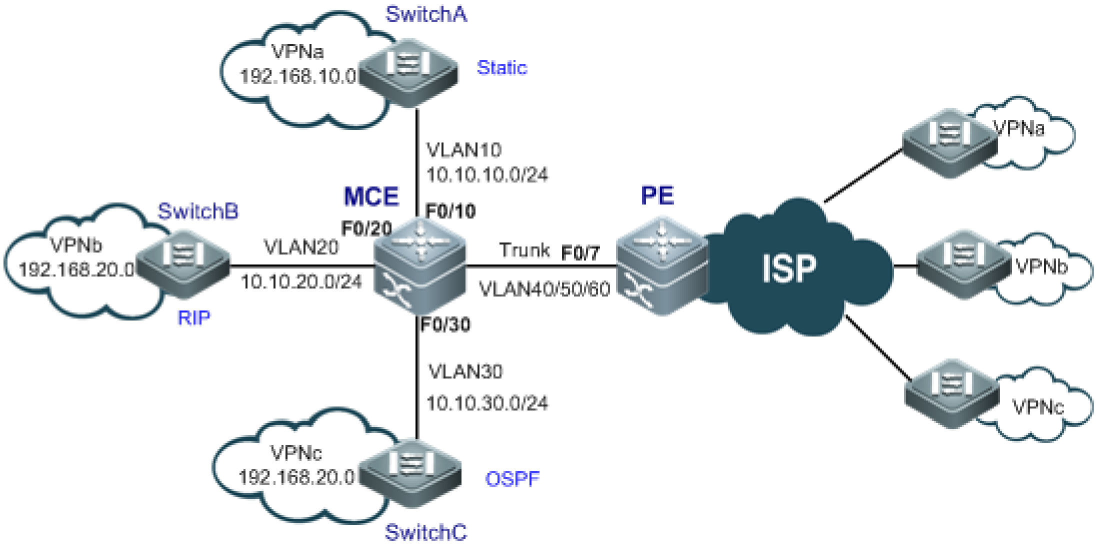
For details about the command syntax, see the *VRF Command Reference*.

MCE Configuration Example

Networking Topology

As shown in the following figure, VPNa, VPNb, and VPNc are located at different sites and exchange information across the backbone network.

- VPN sites access a PE via a MCE.
- Static routes are configured between the MCE and VPNa. RIP is configured between the MCE and VPNb to exchange routing information. The OSPF protocol is configured between the MCE and VPNc to exchange routing information.



Typical Topology of MCE Application

Networking Requirements

The MCE needs to isolate the routes of one VPN from the routes of the other VPNs. It advertises the routes of various VPNs to the PE via the static routing protocol, RIP, and the OSPF protocol respectively.

Duplicate addresses in a VPN must be supported.

Configuration Tips

Create multiple VRF instances on the MCE and the PE, so that the routes of different VPN services are isolated from one another. Perform the following two steps:

Configure VRF instances and bind them to various interfaces.

Interface	Bound VRF	Interface IP Address
Interface of the MCE that connects to VPNa (SVI 10)	VPNa	10.10.10.3
Logical interface of the MCE that connects to the PE (SVI 40)	VPNa	10.10.40.1
Logical interface of the PE that connects to the MCE (SVI 40)	VPNa	10.10.40.2
Interface of the MCE that connects to VPNb (SVI 20)	VPNb	10.10.20.3
Logical interface of the MCE that connects to the PE (SVI 50)	VPNb	10.10.50.1
Logical interface of the PE that connects to the MCE (SVI 50)	VPNb	10.10.50.2
Interface of the MCE that connects to VPNc (SVI 30)	VPNc	10.10.30.3
Logical interface of the MCE that connects to the PE (SVI 60)	VPNc	10.10.60.1
Logical interface of the PE that connects to the MCE (SVI 60)	VPNc	10.10.60.2

Configure route exchange between the MCE, VPN sites, and the PE.

VRF instances on the MCE cooperate with VRF instances on the PE to advertise VPN routes to the PE via the routing protocol between the MCE and the PE. Then the PE advertises the routes to other PEs on the network, so that each VPN site interworks with remote VPN sites on the network.

Note

In this example, VRF instances on the MCE exchange routes with VPN sites and the PE via the same routing protocol.

Note

If VRF instances on the MCE exchange routes with VPN sites and the PE via different routing protocols, to guarantee complete exchange of VPN routes, the routing protocols of VRF instances on the MCE must be redistributed, so that the MCE can advertise VPN routes to VRF instances on the PE and that the routes advertised by the PE to VRF instances on the MCE can be further advertised to VPN sites.

Configuration Steps

Configure VRF instances on the MCE and the PE, and bind VRF instances to interfaces.

Create VRF instances on the MCE.

Step 1: Create VRF instances named “VPNa”, “VPNb”, and “VPNc” on the MCE.

```
MCE(config)#ip vrf vpna
MCE(config-vrf)#exit
MCE(config)#ip vrf vpnb
MCE(config-vrf)#exit
MCE(config)#ip vrf vpnc
MCE(config-vrf)#exit
```

Step 2: Create VLAN 10, and add the FE interface 0/10 of the MCE that connects to Switch A to VLAN 10.

```
MCE(config)#interface fastEthernet 0/10
MCE(config-if-FastEthernet 0/10)#switchport access vlan 10
MCE(config-if-FastEthernet 0/10)#exit
```

Step 3: Bind the interface SVI 10 of VLAN 10 to VPNa, and set the IP address of the interface SVI 10 to 10.10.10.3/24.

```
MCE(config)#interface vlan 10
MCE(config-if-VLAN 10)#ip vrf forwarding vpna
MCE(config-if-VLAN 10)#ip address 10.10.10.3 255.255.255.0
MCE(config-if-VLAN 10)#exit
```

Step 4: Similarly, create VLAN 20 and VLAN 30 (see the previous two steps), bind the interface SVI 20 of the MCE that connects to Switch B to VPNb, and bind the interface SVI 30 of the MCE that connects to Switch C to VPNc. Set the IP address of the interface SVI 20 to 10.10.20.3, and the IP address of the interface SVI 30 to 10.10.30.3.

Step 5: Create VLAN 40, VLAN 50, and VLAN 60, and add the FE interface 0/7 of the MCE that connects to the PE to the three VLANs. Bind SVI 40 to VPNa, SVI 50 to VPNb, and SVI 60 to VPNc. Set the IP address of the interface SVI 40 to 10.10.40.1, the IP address of the interface SVI 50 to 10.10.50.1, and the IP address of the interface SVI 60 to 10.10.60.1.

```
MCE(config)#vlan 40
MCE(config-vlan)#exit
MCE(config)#vlan 50
MCE(config-vlan)#exit
MCE(config)#vlan 60
MCE(config-vlan)#exit
MCE(config)#interface fastEthernet 0/7
MCE(config-if-FastEthernet 0/7)#switchport mode trunk
MCE(config-if-FastEthernet 0/7)#exit
MCE(config)#interface vlan 40
MCE(config-if-VLAN 40)#ip vrf forwarding vpna
MCE(config-if-VLAN 40)#ip address 10.10.40.1 255.255.255.0
MCE(config-if-VLAN 40)#exit
MCE(config)#interface vlan 50
MCE(config-if-VLAN 50)#ip vrf forwarding vpnb
MCE(config-if-VLAN 50)#ip address 10.10.50.1 255.255.255.0
MCE(config-if-VLAN 50)#exit
MCE(config)#interface vlan 60
MCE(config-if-VLAN 60)#ip vrf forwarding vpnc
MCE(config-if-VLAN 60)#ip address 10.10.60.1 255.255.255.0
```

After the above steps are performed, VRF instances are created on the MCE.

Create VRF instances on the PE.

Step 1: Create VRF instances named “VPNa”, “VPNb”, and “VPNc” on the PE.

```
PE(config)#ip vrf vpna
PE(config-vrf)#exit
PE(config)#ip vrf vpnb
PE(config-vrf)#exit
PE(config)#ip vrf vpnc
PE(config-vrf)# exit
```

Step 2: Create VLAN 40, VLAN 50, and VLAN 60, and add the FE interface 0/7 of the PE that connects to the MCE to the three VLANs. Bind SVI 40 to VPNa, SVI 50 to VPNb, and SVI 60 to VPNc. Here, the operations for creating VLANs and adding interfaces to VLANs are omitted.

```
PE (config)#interface vlan 40
PE(config-if-VLAN 40)#ip vrf forwarding vpna
PE(config-if-VLAN 40)#ip address 10.10.40.2 255.255.255.0
PE(config-if-VLAN 40)#exit
PE(config)#interface vlan 50
PE(config-if-VLAN 50)#ip vrf forwarding vpnb
PE(config-if-VLAN 50)#ip address 10.10.50.2 255.255.255.0
PE(config-if-VLAN 50)#exit
PE(config)#interface vlan 60
PE(config-if-VLAN 60)#ip vrf forwarding vpnc
PE(config-if-VLAN 60)#ip address 10.10.60.2 255.255.255.0
```

After the above steps are performed, VRF instances are created on the PE.

Configure static routes between the MCE and VPNa, and between the MCE and the PE.

Configure a static route on Switch A (access switch for the site VPNa).

Set the IP address of the interface of Switch A that connects to the MCE to 10.10.10.2/24, and that of the interface of Switch A that connects to the site VPNa to 192.168.10.1/24. Here, the operations for adding ports to VLANs and setting interface IP addresses are omitted.

# Configure a default route on Switch A, with the next hop of the outbound packet pointing to 10.10.10.3.

```
SwitchA(config)#ip route 0.0.0.0 0.0.0.0 10.10.10.3
```

Configure a static route on the MCE.

# Configure a static route on the MCE for packets destined to the network segment 192.168.10.0, with the next hop pointing to 10.10.10.2, and binds the route to the instance VPNa.

```
MCE(config)#ip route vrf vpna 192.168.10.0 255.255.255.0 10.10.10.2
```

Configure static routes on the PE.

# Configure two static routes on the PE, and bind them to the instance VPNa. One is for packets destined to the network segment 192.168.10.0, with the next hop pointing to 10.10.40.1. The other is for packets destined to the network segment 10.10.10.0, with the next hop also pointing to 10.10.40.1.

```
PE (config)#ip route vrf vpna 192.168.10.0 255.255.255.0 10.10.40.1
PE (config)#ip route vrf vpna 10.10.10.0 255.255.255.0 10.10.40.1
```

Configure RIP route exchange between the MCE and VPNb, and between the MCE and the PE.

Configure RIP on Switch B (access switch for the site VPNb).

# Set the IP address of the interface of Switch B that connects to the MCE to 10.10.20.2/24, and that of the interface of Switch B that connects to the site VPNb to 192.168.20.1/24. Here, the operations for adding ports to VLANs and setting interface IP addresses are omitted.

```
SwitchB(config)#router rip
SwitchB(config-router)#version 2
SwitchB(config-router)#no auto-summary
SwitchB(config-router)#network 10.10.20.0 0.0.0.255
SwitchB(config-router)#network 192.168.20.0 0.0.0.255
```

Configure RIP on the MCE.

```
MCE(config)#router rip
MCE(config-router)#address-family ipv4 vrf vpnb
MCE(config-router-af)# version 2
MCE(config-router-af)# no auto-summary
MCE(config-router-af)#network 10.10.20.0 0.0.0.255
MCE(config-router-af)#network 10.10.50.0 0.0.0.255
```

Configure RIP on the PE.

```
PE(config)#router rip
PE(config-router)#address-family ipv4 vrf vpnb
PE(config-router-af)# version 2
PE(config-router-af)# no auto-summary
```

PE(config-router-af)#network 10.10.50.0 0.0.0.255 Configure OSPF route exchange between the MCE and VPNc, and between the MCE and the PE.

Configure the OSPF protocol on Switch C (access switch for the site VPNc).

# Set the IP address of the interface of Switch C that connects to the MCE to 10.10.30.2/24, and that of the interface of Switch C that connects to the site VPNc to 192.168.20.1/24. Here, the operations for adding ports to VLANs and setting interface IP addresses are omitted.

```
SwitchC(config)#router ospf 1
SwitchC(config-router)#network 10.10.30.0 0.0.0.255 area 0
SwitchC(config-router)#network 192.168.20.0 0.0.0.255 area 0
```

Configure the OSPF protocol on the MCE.

```
MCE(config)#router ospf 1 vrf vpnc
MCE(config-router)#network 10.10.30.0 0.0.0.255 area 0
MCE(config-router)#network 10.10.60.0 0.0.0.255 area 0
```

Configure the OSPF protocol on the PE.

```
PE(config)#router ospf 1 vrf vpnc
PE(config-router)#network 10.10.60.0 0.0.0.255 area 0
```

Verification

Check routing information about the instance VPNa.

Check routing information on Switch A (access switch for the site VPNa).

```
SwitchA (config)#show ip route
Gateway of last resort is 10.10.10.3 to network 0.0.0.0
S*  0.0.0.0/0 [1/0] via 10.10.10.3
C   10.10.10.0/24 is directly connected, VLAN 10
C   10.10.10.2/32 is local host.
C   192.168.10.0/24 is directly connected, FastEthernet 0/23
C   192.168.10.1/32 is local host.
```

Check routing information about the instance VPNa on the MCE.

```
MCE#show ip route vrf vpna
Routing Table: vpna
C   10.10.10.0/24 is directly connected, VLAN 10
C   10.10.10.3/32 is local host.
C   10.10.40.0/24 is directly connected, VLAN 40
C   10.10.40.1/32 is local host.
S   192.168.10.0/24 [1/0] via 10.10.10.2
```

Check routing information about the instance VPNa on the PE.

PE#show ip route vrf vpna

```
Routing Table: vpna
S   10.10.10.0/24 [1/0] via 10.10.40.1
C   10.10.40.0/24 is directly connected, VLAN 40
C   10.10.40.2/32 is local host.
S   192.168.10.0/24 [1/0] via 10.10.40.1
```

Check routing information about the instance VPNb.

Check routing information on Switch B (access switch for the site VPNb).

SwitchB#show ip route vrf vpnb

```
Routing Table: vpnb
C   10.10.20.0/24 is directly connected, VLAN 20
C   10.10.20.2/32 is local host.
R   10.10.50.0/24 [120/1] via 10.10.20.3, 00:01:20, VLAN 20
C   192.168.20.0/24 is directly connected, FastEthernet 0/23
C   192.168.20.1/32 is local host.
```

Check routing information about the instance VPNb on the MCE.

```
MCE#show ip route vrf vpnb
Routing Table: vpnb
C   10.10.20.0/24 is directly connected, VLAN 20
```

```
C    10.10.20.3/32 is local host.
C    10.10.50.0/24 is directly connected, VLAN 50
C    10.10.50.1/32 is local host.
R    192.168.20.0/24 [120/1] via 10.10.20.2, 00:22:01, VLAN 20
```

According to the above information, the MCE has learned the private network routes of VPNb via RIP. These routes and the routes of VPNa and VPNc are separately maintained in three routing tables, thus effectively isolating the VPNs from each other and supporting address duplication inside each VPN.

Check routing information about the instance VPNb on the PE.

```
PE#show ip route vrf vpnb
Routing Table: vpnb
R    10.10.20.0/24 [120/1] via 10.10.50.1, 00:04:48, VLAN 50
C    10.10.50.0/24 is directly connected, VLAN 50
C    10.10.50.2/32 is local host.
R    192.168.20.0/24 [120/2] via 10.10.50.1, 00:02:15, VLAN 50
```

The above information shows that all the routes of the instance VPNb have been advertised to the PE.

Check routing information about the instance VPNc.

Check routing information on Switch C (access switch for the site VPNc).

```
SwitchC (config-router)#show ip route

C    10.10.30.0/24 is directly connected, VLAN 30
C    10.10.30.2/32 is local host.
O    10.10.60.0/24 [110/2] via 10.10.30.3, 00:02:42, VLAN 30
C    192.168.20.0/24 is directly connected, FastEthernet 0/23
C    192.168.20.1/32 is local host.
```

Check routing information about the instance VPNc on the MCE.

```
MCE#show ip route vrf vpnc

Routing Table: vpnc
C    10.10.30.0/24 is directly connected, VLAN 30
C    10.10.30.3/32 is local host.
C    10.10.60.0/24 is directly connected, VLAN 60
C    10.10.60.1/32 is local host.
O    192.168.20.0/24 [110/2] via 10.10.30.2, 00:01:36, VLAN 30
```

According to the above information, the MCE has learned the private network routes of VPNc via OSPF. These routes and the routes of VPNa and VPNb are separately maintained in three routing tables, thus effectively isolating the VPNs from each other and supporting address duplication inside each VPN.

Check routing information about the instance VPNc on the PE.

```
PE#show ip route vrf vpnc

```

```
Routing Table: vpnc
O    10.10.30.0/24 [110/2] via 10.10.60.1, 00:00:00, VLAN 60
C    10.10.60.0/24 is directly connected, VLAN 60
C    10.10.60.2/32 is local host.
O    192.168.20.0/24 [110/3] via 10.10.60.1, 00:00:00, VLAN 60
```

The above information shows that all the routes of the instance VPNc have been advertised to the PE.

Abbreviations

Abbreviation	Full Spelling
CE	Customer Edge Device
PE	Provider Edge Device
MCE	Multi-CE
VPN	Virtual Private Network
VRF	VPN Routing and Forwarding Table

Configuring IPv4 Express Forwarding

Understanding IPv4 Express Forwarding

Overview

To meet the needs of higher-end devices, we use a Prefix Tree+Adjacency Ruijie Express Forwarding (REF) model to enable express forwarding. REF constitutes the mirroring of the whole core routing table instead of buffering some information in the core routing table. Therefore, the cache does no longer need to be added to the CPU in the case of a cache failure, thus reducing the impact on the CPU and guaranteeing routing stability.

REF constructs the routing table mirroring by using the following two parts:

- Prefix Tree

The Prefix Tree is an IP prefix tree organized according to the maximum matching rule for retrieving adjacent nodes. In general, the data structure for constructing the Prefix Tree is different from the Radix Tree of the core routing table. Instead, a data structure called the M-Tries Tree is used to enable more rapid multi-step search. Using the M-Tries Tree to construct the Prefix Tree will take up more memory than using the Radix Tree. Although updating prefix and node information relatively consumes time, high retrieval performance can be obtained.

- Adjacency

Adjacency refers to an adjacent node, which contains the interface information output by routed packets, such as next-hop list, next processing part, and link-layer output encapsulation. When packets match with such an adjacent node, packets are directly encapsulated, and then the send function of this node is called to enable forwarding. To facilitate retrieval and update, the tables made up of adjacent nodes are generally organized into a hash table. To support load balancing of routes, the next list information on adjacent nodes is organized in load-balancing table form.

REF routing consists of three steps:

- REF de-encapsulates packets.
- Use the Prefix Tree to retrieve the adjacent nodes according to the routes of packets.
- After the adjacent nodes are matched, the final output interface of packets is determined according to adjacent nodes, and then packets are encapsulated according to the output interface type.

Configuring Load Balancing Policy for Express Forwarding

Express forwarding supports load balancing of packets. At present, three load balancing policies are supported. In the REF model, when the route prefix IP/MASK is associated with multiple next hops, namely, a multi-path route, the route will be associated with one load-balancing table and load is balanced according to route weights. When IP packets match with the load balancing table according to the longest prefix, express forwarding hashes the IP addresses of packets, and then one path is selected for packet forwarding. There arethree routing policies:

- Load is balanced according to destination addresses of IP packets. The destination IP addresses of packets are hashed. The route with a greater weight is more likely to be selected. By default, this policy is selected.
- Load is balanced according to destination and source IP addresses of IP packets. Destination and source IP addresses of packets are hashed and the route with a greater weight is more likely to be selected.
- Load is balanced according to packets polling. Each packet takes turn to select the path and all paths can be selected.

Use the following commands to configure the load balancing policy in global configuration mode:

Command	Function
Ruijie(config)# ip ref load-sharing algorithm original	Sets the load balancing algorithm based on source and destination IP addresses.
Ruijie(config)# **ip ref load-sharing packet**	Performs the load balancing according to packet polling
Ruijie(config)# no ip ref load-sharing algorithm	Load is balanced according to destination addresses of packets.

Note These commands are unique to routers.

Maintaining and Monitoring the Express Forwarding Table

The express forwarding module passively receives and maintains external routing information instead of actively adding or deleting any route information. As a result, the express forwarding table provides current statistical information on routes.

Statistical information on Express Forwarding Packets

Statistical information on express forwarding packets refers to the statistical information on the packets processed by the express forwarding REF, including the number of forwarded packets and the number of packets dropped due to various reasons.

Command	Function
Ruijie# show ip ref packet-statistic [clear]	Displays or clears the current statistical information on packets in REF.

Adjacency Table Information

In the express forwarding table, one type of important data table is the adjacency table. Use the following command to check current adjacency information.

Command	Function

Ruijie# show ip ref adjacency [glean | local | *ip* | (interface *interface_type* *interface_number*) | statistic]	Displays the specified gleaned adjacency, local adjacency, adjacency corresponding to the specified IP address, adjacency associated with the specified interface, and information on all adjacency nodes.

Information on Packet Forwarding Path

Packets are forwarded according to IP addresses of packets. Therefore, if source IP addresses and destination IP addresses of packets are specified, the path for forwarding packets will be determined. Use the following command and specify source IP addresses and destination IP addresses of packets. The actual path for forwarding packets will be displayed. For example, you can learn whether packets are dropped, submitted to the CPU, or forwarded. Furthermore, you can know about the interface through which packets are forwarded.

Command	**Function**
Ruijie# show ip ref exact-route [vrf *vrf_name*] *source-ipaddress dest_ipaddress*	Displays the actual forwarding path of a specified packet.

Note This command is unique to routers.

Routing information in the Express Forwarding Table

Express forwarding receives external route advertisements and maintains an express forwarding table, which is a mirroring of the core routing table. Use the following commands to display routing information in the express forwarding table.

Command	**Function**
Ruijie# show ip ref route [vrf vrf_name] [default | (ip mask) | statistic]	Displays current default routing information in the express forwarding table. If the default route is not specified, all routing information in the express forwarding table is displayed, including the 0 route, default route, and common gateway route.

Configuring TCP

Overview

The TCP module provides a reliable and connection-oriented IP-based transmission layer protocol for the application layer.

The application layer sends data streams represented in 8-bit bytes for Internet transmission to the TCP layer, which separates the data streams into packet segments with proper sizes. The maximum segment size (MSS) is generally limited by the maximum transmission unit (MTU) of the data link layer of the network to which the computer is connected. After that, TCP transmits the result packets to the IP layer, which will then transmit the said packets through the network to the TCP layer of receiving terminal.

To ensure no packet loss, TCP assigns a sequence number to each byte, and the sequence number also ensures that packets transmitted to the receiving terminal are received in sequence. The receiving terminal will then reply with an ACK to confirm the receipt of each byte. If no ACK is received within the reasonable Round Trip Time (RTT), then the corresponding byte (assumed lost) will be retransmitted by the sender.

- With regard to data accuracy and validity, TCP uses a checksum function to verify the data. The checksum must be calculated while the date is sent or received. In the meantime, MD5 authentication can also be utilized to encrypt the data.
- To ensure reliability, TCP applies the mechanisms of timeout retransmission and piggybacking.
- The sliding window protocol is applied to implement flow control. According to the protocol, all unconfirmed packets within the window will be retransmitted.
- The widely recognized TCP congestion control algorithm (also called the AIMD algorithm) is applied to implement congestion control. This algorithm mainly involves: 1) additive increase and multiplicative decrease; 2) slow start; 3) response to timeouts.

Configuring TCP

Changing the Timeout for Establishing a TCP Connection

Establishing a TCP connection requires a three-way handshake: The local end sends a SYN packet, the remote end responds with a SYN+ACK packet, and then the local end responds with an ACK.

- After the local end sends SYN, if the remote end does not respond with SYN+ACK, the local end will continuously retransmit SYN packets until the specified number of retransmissions is reached or until the timeout timer expires.
- After the local end sends SYN and the remote end responds with SYN+ACK, if the local end no longer responds with ACK, the remote end will keep retransmission until the specified number of retransmissions is reached or until the timeout timer expires (Such as SYN attacks).

Use the following command to configure the timeout value for SYN packets (the maximum time from SYN transmission to successful three-way handshake), namely the timeout for establishing a TCP connection.

Command	Function
Ruijie(config)# ip tcp syntime-out *seconds*	Changes the timeout value for establishing a TCP connection. Range: 5 to 300 seconds; default: 20

Use the **no ip tcp syntime-out** command to restore the default value.

Note This command only applies to IPv4 TCP.

Changing the Buffer Size

The TCP receiving buffer is utilized to buffer the data received from the peer end. These data will be subsequently read by application programs. Generally, the window size of TCP packets implies the size of the free space in the receiving buffer. For connections involving a greater bandwidth and mass data, increasing the size of the receiving buffer will remarkably improve TCP transmission performance. The sending buffer is utilized to buffer the data of application programs. Each byte in the buffer has a sequence number, and bytes with sequence numbers acknowledged will be removed from the sending buffer. Increasing the sending buffer will improve the interaction between TCP and application programs, thus enhancing the performance. However, increasing the receiving buffer and sending buffer will result in more memory consumption of TCP.

Command	Function
Ruijie(config)# ip tcp window-size *size*	Changes the size of receiving buffer and sending buffer for TCP connections. Range: 0 to 65535 bytes; default: 4096.

Use the **no ip tcp window-size** command to restore the default value.

Note This command only applies to IPv4 TCP.

Note This command does not apply to existing TCP connections. It only applies to subsequent TCP connections.

Note This command will apply to both the receiving buffer and sending buffer.

Prohibiting the Reset Packet When the Port is Unreachable

When the TCP module distributes TCP packets, if the TCP connection to which such packets belong cannot be found, a reset packet will be replied to the peer end to terminate the TCP connection. The attacker may initiate attacks by sending excessive port-unreachable TCP packets.

Run the following commands to prohibit or restore the reset packet sent when the port-unreachable TCP packet is received.

Command	Function
Ruijie(config)# ip tcp not-send-rst	Prohibits sending a reset packet when the port-unreachable TCP packet is received.

Use the **no ip tcp not-send-rst** command to restore the default settings.

Note

This command only applies to IPv4 TCP.

Limiting the MSS of TCP Connections

The MSS refers to the maximum size of the payload of a TCP packet, excluding TCP options.

During the three-way handshake for establishing a TCP connection, one important job is to carry out MSS negotiation. Both sides will insert an MSS option into the SYN packet to indicate the maximum size of the segment that can be received by the local end, namely the maximum size of the segment that can be sent by the remote end. Both sides will take the smaller of the MSS value sent locally and that received from the remote end as the maximum segment size of this connection.

The methods for calculating the value of the MSS option while sending SYN packets are shown below:

Non-directly connected network: MSS = Default value 536

Directly connected network: mss = egress interface MTU corresponding to the peer IP address – 20-byte IP header – 20-byte TCP header

Generally speaking, if the MTU is affected by certain applications configured on the egress interface, such applications will configure the MTU accordingly, such as the MTUs of the tunnel port and VPN port.

Note

In release 10.4(3), in the SYN+ACK packet replied by the remote end of a directly connected network, the MSS option is not calculated through MTU. Instead, the default value 536 is used.

Note

The MSS calculated cannot exceed the size of the receiving buffer or the IP TCP MSS configured by the user. Otherwise, the smaller of them will be used.

Note

If certain options are supported by this connection, then the size obtained after 4-byte alignment of the option must be subtracted from the MSS. For example, if the size of the MD5 option is 18 bytes, 20 bytes will be obtained after alignment.

The RMSS value obtained here is the value of the MSS option in the SYN packet sent. For example, BGP adjacency is generally established in the directly connected network, and the MSS of such a connection is 1500 – 20 – 20 – 20 = 1440.

The function of IP TCP MSS is to limit the MSS of the pending TCP connection. The negotiated MSS cannot exceed the value configured.

Command	Function
Ruijie(config)# **ip tcp mss** *max-segment-size*	Limits the maximum segment size of TCP connections. Range: 68 to 10000 bytes.

Use the **no ip tcp mss** command to disable such limit.

Note This command only applies to IPv4 TCP.

Enabling PMTU Discovery

The TCP Path MTU (PMTU) is implemented as per RFC1191. This feature can improve the network bandwidth utilization ratio. When the user uses TCP to transmit mass data, this feature can substantially enhance the transmission performance.

Command	Function
Ruijie(config)# **ip tcp path-mtu-discovery** [**age-timer** *minutes* | **age-timer infinite**]	Enables PMTU discovery. **age-timer** *minutes*: The time interval for further discovery after discovering PMTU in the range from 10 to 30 minutes. The default value is 10. **age-timer infinite**: No further discovery after discovering PMTU.

According to RFC1191, after discovering PMTU, TCP can use a greater MSS to discover a new PMTU, and the time interval thereof is specified by the parameter **age-timer**. When the PMTU discovered by the device is smaller than the MSS negotiated, the device will try to discover a greater PMTU as per the aforementioned time interval. Such a discovery process will not end until PMTU reaches the value of MSS or until the user stops this timer. To turn off the timer, use the parameter **age-timer infinite**.

Use the **no ip tcp path-mtu-discovery** command to disable PMTU discovery.

Note This command applies to both IPv4 TCP and IPv6 TCP.

Note This command does not apply to existing TCP connections. It only applies to subsequent TCP connections.

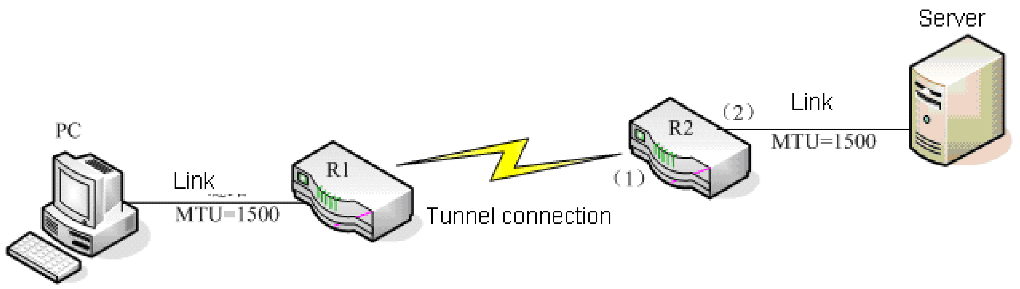
Configuring the MSS Option of SYN Packets Sent and Received on the Interface

The TCP Path MTU (PMTU) is implemented as per RFC1191. This feature can improve the network bandwidth utilization ratio. When the user uses TCP to transmit mass data, this feature can substantially enhance the transmission performance.

When the client initiates a TCP connection, it negotiates the maximum payload of TCP packets through the MSS Option field of the TCP SYN packet. The MSS value of client's SYN packet implies the maximum payload of TCP packets sent by the server, and vice versa.

As shown in the following figure, a PC may fail to access the server through HTTP, because the MSS of 1460 will be negotiated between the PC and the server, but such MSS cannot pass R1 and R2 (R1 and R2 are connected through a tunnel, with an MTU smaller than 1500).

Figure 1-1



In such a case, we can configure the following command on port (1) and port (2) of R2 to change the MSS Option value of the SYN packet, so as to change the MSS value negotiated for the TCP connection going through port (1) and port (2).

Command	Function
Ruijie(config-if)# **ip tcp adjust-mss** *max-segment-size*	Configures the MSS Option value of SYN packets sent and received on the interface. Range: 500 to 1460 bytes.

Use the **no ip tcp adjust-mss** command to remove the configuration. In such a case, the MSS Option value of packets will not be changed when the interface sends and receives SYN packets.

Configuring this command on the interface will change the MSS option of SYN packets received or sent on the interface to the MSS value configured on the interface. It is suggested that you configure the same value on the ingress interface and egress interface, or else the MSS option of SYN packets going through the device will be changed to the smaller of the two values configured.

Note

This command only applies to IPv4 TCP and is only supported by routers.

Note

In release 10.4(3), the MSS value of SYN+ACK packets will not be changed.

Monitoring and Maintenance

Command	Function
Ruijie# show tcp connect	Displays basic information about the current TCP connection.
Ruijie# show tcp pmtu	Displays information about TCP PMTU.
Ruijie# show tcp port	Displays information about the current TCP port.

Application Protocol Configuration

Configuring DNS

Overview

Due to the Domain Name System (DNS), each IP address can present a host name consisting of one or more strings separated by the decimal. Then, you only need to remember the host name rather than IP address.

There are two methods of mapping the host name to the IP address: 1) Static mapping: A device maintains its host name to IP address mapping table and uses it only by itself. 2) Dynamic mapping: The host name to IP address mapping table is maintained on the DNS server. In order for a device to communicate with others by its host name, the corresponding IP address needs to be searched on the DNS server.

The domain name resolution (or host name resolution) is the process that the device obtains an IP address corresponding to the host name by the host name. Ruijie switches support the host name resolution locally or by the DNS. During domain name resolution, you can firstly adopt the static method. If it fails, use the dynamic method. Some frequently used domain names can be put into the resolution list of static domain names. In this way, the efficiency of domain name resolution can be increased considerably.

Configuring Domain Name Resolution

Default DNS Configuration

The following table describes default configurations of DNS.

Attribute	**Default value**
Enable/disable the DNS resolution service	Enable
IP address of DNS server	None
Status Host List	None
Maximum number of DNS servers	Six

Enabling DNS Resolution Service

The following table describes how to enable the DNS resolution service.

Command	**Function**
Ruijie(config)# **ip domain**-lookup	Enables DNS.

To disable DNS, use the **no ip domain-lookup** command.

```
Ruijie(config)# ip domain-lookup
```

Configuring the DNS Server

This section describes how to configure the DNS server. The dynamic domain name resolution can be carried out only when the DNS Server is configured.

Use the **no ip name-server** [*ip-address*] command to remove the DNS server. In the command, the *ip-address* parameter indicates the specified DNS server to be removed. If this parameter is omitted, all the DNS servers will be removed.

Command	Function
Ruijie(config)**# ip name-server** *ip-address*	Adds the IP address of the DNS Server. The switch will add a DNS Server every time this command is executed. If the domain name can't be obtained from the first DNS Server, the switch will send the DNS request to the subsequent several servers until the correct response is received. The system can support six DNS servers at most.

Configuring the Host Name to IP/IPv6 Address Mapping Statically

This section describes how to configure the host name to IP/IPv6 address mapping. The switch maintains a host name to IP/IPv6 address corresponding table, which is also referred to as the host name to IP/IPv6 address mapping table. You can obtain the mapping table in two ways: manual configuration and dynamic learning. Manual configuration is required when dynamic learning is impossible.

Command	Function
Ruijie(config)# **ip host** *host-name ip-address*	Configures the host name to IP address mapping manually.
Ruijie(config)# **ipv6 host** *host-name ip-address*	Configures the host name to IPv6 address mapping manually.

To remove the mapping between the host name and IP/IPv6 address, use the **no** form of this command.

Clearing the Dynamic Buffer Table of Host Names

This section describes how to clear the dynamic buffer table of host names. If the **clear host** or **clear host** * command is entered, the dynamic buffer table will be cleared. Otherwise, only the entries of specified domain names will be cleared.

The following table describes related command.

Command	Function
Ruijie# **clear host** [*word*]	Clears the dynamic buffer table of host names. The host names configured statically will not be removed.

Showing Domain Name Resolution Information

This section describes how to display the DNS configuration.

Command	Function

Command	Function
Ruijie# **show hosts** [*host-name*]	Show parameters related to DNS.

```
Ruijie# show hosts
Name servers are:
192.168.5.134  static
Host                 type      Address          TTL(sec)
www.163.com          static    192.168.5.243    ---
```

Typical DNS Configuration Examples

Example of Static DNS Configuration

Topological Diagram

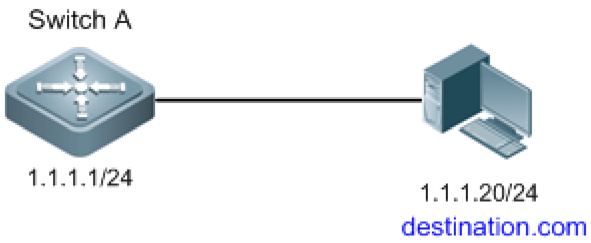


Figure1 Networking topology for static DNS configuration

Application Requirements

Since Switch A will frequently access the host of destination.com, you can use static DNS to access the host of IP 1.1.1.20 through the domain name of destination.com for domain resolution efficiency.

Configuration Tips

- Ensure that the route between device and host is reachable.
- The mapping between host name and IP address is correct.

Configuration Steps

Manually configure the mapping between host name and IP address. In this example, configure the host name to "destination.com" and the corresponding IP address to 1.1.1.20.

```
SwitchA(config)#ip host destination.com 1.1.1.20
```

Verification

Step 1: Show DNS information. Key points: the mapping between host and IP address shall be correct.

```
Ruijie-A# show host
```

```
Name servers are:
Host               type    Address                   TTL(sec)
destination.com    static  1.1.1.20                   ---
```

Step 2: Use the **ping destination.com** command to verify the result.

```
Ruijie-A# ping destination.com
Translating "destination.com"...[OK]
Sending 5, 100-byte ICMP Echoes to 1.1.1.20, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

The output shows that Ruijie-A has successfully accessed the host with IP address being 1.1.1.20 through the host name of destination.com by means of static DNS.

Example of Dynamic DNS Configuration

Topological Diagram

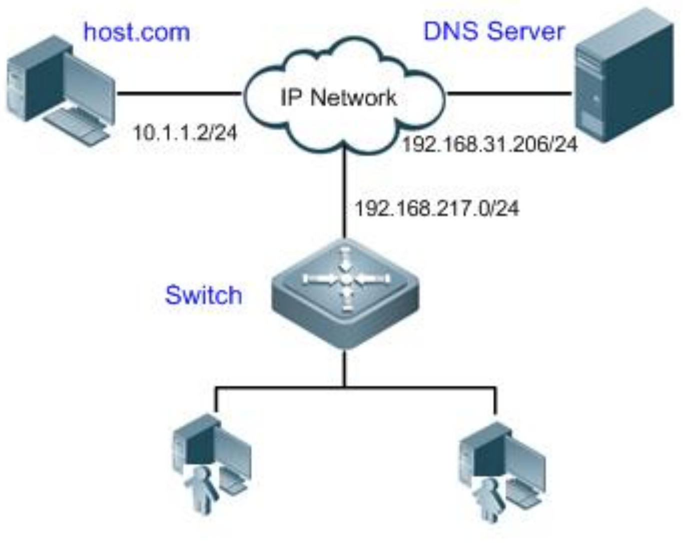


Figure2 Networking topology for dynamic DNS configuration

Application Requirements

- The IP address of DNS server is 192.168.31.206/24.
- The switch is the DNS client that can access the host of 10.1.1.2 through the host name of host.com by means of dynamic DNS.

Configuration Tips

- The route between DNS client, DNS server, and access PC shall be reachable.
- DNS shall be enabled. DNS is enabled by default.
- The IP address of the DNS server has been correctly configured.

Configuration Steps

Step 1: Configure the DNS server

Configure different DNS servers according to the actual conditions.

Configure the mapping between host and IP address on DNS server. This example configures the host name as "host.com" and the IP address as 10.1.1.2/24.

Step 2: Configure the DNS client

The route between DNS client, DNS server, and access PC shall be reachable. The interface IP configurations are shown in the topological diagram.

! DNS shall be enabled. The DNS feature is enabled by default.

```
Ruijie(config)#ip domain-lookup
```

! Configure the IP address of DNS server as 192.168.31.206

```
Ruijie(config)#ip name-server 192.168.31.206
```

Verification

Step 1: Use the **ping host.com** command to verify the result.

```
Ruijie#ping host.com

Translating " host.com "...[OK]
Sending 5, 100-byte ICMP Echoes to 10.1.1.2, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
```

The preceding information shows that the client can ping the host, and the destination IP is 10.1.1.2. Through dynamic DNS, the host with the IP address 10.1.1.2 can be accessed through the host name of host.com.

Step 2: Show DNS information. Key points: the host name and IP address.

```
Ruijie#show host
Name servers are:
192.168.31.206 static

Host               type      Address         TTL(sec)
host.com          dynamic    10.1.1.2         3503
```

The output shows that the mapping between host name and host IP is correct.

Configuring DHCP

Understanding DHCP

The Dynamic Host Configuration Protocol (DHCP), as specified in RFC 2131, provides configuration parameters for hosts over the Internet. DHCP works in client/server mode. The DHCP server assigns IP addresses for the hosts dynamically and provides configuration parameters.

DHCP assigns IP address in three ways:

- Assigning IP addresses automatically. The DHCP server assigns permanent IP addresses to the clients;
- Assigning IP addresses dynamically. The DHCP server assigns temporary IP addresses to the clients (or the clients can release the addresses by themselves);
- Configuring IP addresses manually. Network administrators specify IP addresses and send the specified IP addresses to the clients through DHCP.

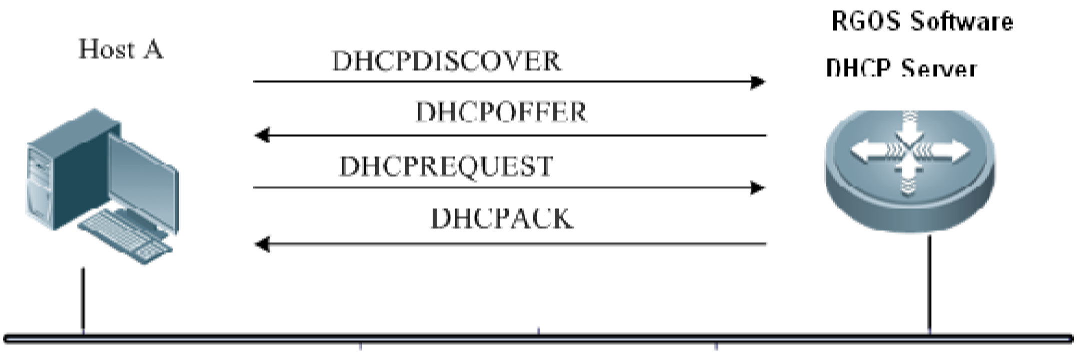
Among the mentioned three methods, only dynamic assignment allows reuse of the IP address that the client does not need any more.

The format of a DHCP message is based on that of a Bootstrap Protocol (BOOTP) message. Hence, the device must act as the BOOTP relay agent and interact with the BOOTP client and the DHCP server. The BOOTP relay agent eliminates the need of deploying a DHCP server in every physical network. DHCP is detailed in RFC 2131 and RFC 2132.

Understanding the DHCP Server

As specified in RFC2131, the DHCP server of Ruijie Networks is implemented to assign and manage IP addresses for the DHCP clients. The DHCP operation process is shown in the following figure.

Fig 1-1 DHCP process



Process of requesting an IP address:

The host broadcasts a DHCPDISCOVER packet in the network to locate the DHCP server;
The DHCP server sends a DHCPOFFER packet in unicast form to the host, including IP address, MAC address, domain name and address lease period;
The host sends a DHCPREQUEST packet in broadcast form to formally request the server to assign the provided IP address;

The DHCP server sends a DHCPACK packet in unicast form to the host to confirm the request.

Note

The DHCP client may receive the DHCPOFFER packets from multiple DHCP servers, and accept all received DHCPOFFER packet. However, the DHCP client usually accepts the first received DHCPOFFER packet only.

Note

The address specified in the DHCPOFFER packet from the DHCP server is unnecessarily the finally assigned address. Generally, the DHCP server reserves this address until the client sends a formal request.

The DHCPREQUEST that requests the DHCP server to assign an address is a broadcast packet with the server address to enable all other DHCP servers that send DHCPOFFER response packets to receive the packet. Other DHCP servers are unable to find that the client has received the DHCPOFFER packet from just the DHCPREQUEST packet, so they will not release the IP addresses provided (pre-assigned) to the clients and will enable the IP addresses corresponding to the unaccepted OFFER lease to be reused through the timing mechanism.

If the DHCPOFFER packet sent to the DHCP client contains invalid parameters, the DHCP client sends the DHCPDECLINE packet to refuse the assigned configuration.

The following are advantages of using the DHCP server of Ruijie Networks for network construction:

- Decreases network access cost. Generally, dynamic address assignment costs less than static address assignment.
- Simplifies configuration tasks and reduce network construction cost. Dynamic address assignment significantly simplifies equipment configuration, and even reduces deployment cost if devices are deployed in the places where there are no professionals.
- Centralizes management. During configuration management on several subnets, each configuration parameter can be changed simply by modifying and updating configurations in the DHCP server.

Understanding the DHCP Client

The DHCP client can obtain IP addresses and other configuration parameters from the DHCP server automatically. The DHCP client brings the following advantages:

- Saves device configuration and deployment time.
- Reduces the possibility of configuration errors.
- Centrally manages IP address assignment.

☑ The DHCP client is supported on the Ethernet interface, FR, PPP, and HDLC interfaces.

Understanding the DHCP Relay Agent

The DHCP relay agent forwards DHCP packets between the DHCP server and the DHCP clients. When the DHCP clients and the server are not located in the same subnet, the DHCP relay agent must be available for forwarding the DHCP request and response messages. Data forwarded by the DHCP relay agent is different from general forwarding. In general forwarding, IP packets are unaltered and the transmission is transparent. However, upon receiving a DHCP message, the DHCP relay agent regenerates a DHCP message before forwarding it.

For the DHCP client, the DHCP relay agent works like a DHCP server. For the DHCP server, the DHCP relay agent works like a DHCP client.

Configuring DHCP

To configure DHCP, perform the following tasks, of which the first three tasks are mandatory.

Enabling the DHCP Server and Relay Agent

Use the following commands to enable the DHCP server and relay agent in global configuration mode.

Command	Function
Ruijie(config)# **service dhcp**	Enables the DHCP server and the DHCP relay agent.
Ruijie(config)# **no service dhcp**	Disables the DHCP server and the DHCP relay agent.

☑ By default, in v10.1 and later, the **service dhcp** command can be used for both the DHCP server and DHCP relay, which are two mutually-exclusive functions. The switchover of those two functions depends on whether the DHCP address pool is configured.

☑ However, for the product in the version earlier than v10.1 (excluding v10.1), the **service dhcp** command is not supported by both DHCP server and DHCP relay. You can use the **service dhcp** command to enable the DHCP service or the DHCP server.

☑ For some products in v10.1 and later, DHCP may conflict with some functions. For more information, see the prompting message of a specific product.

Configuring DHCP Excluded Addresses

Unless configured particularly, the DHCP server tries to assign all the subnet addresses defined in the address pool to the DHCP clients. To reserve some addresses, such as those addresses that have been assigned, you must define those addresses as excluded.

Use the following commands to configure the excluded addresses in global configuration mode.

Command	Function
Ruijie(config)# **ip dhcp excluded-address** *low-ip-address [high-ip-address]*	Defines excluded addresses.
Ruijie(config)# **no ip dhcp excluded-address** *low-ip-address [high-ip-address]*	Removes the configuration.

Note

A good practice in configuring the DHCP server is to prohibit the DHCP server from assigning any address that has been assigned specifically. This provides two advantages: 1) No address conflict will occur; 2) When DHCP assigns addresses, the time for detection is shortened and DHCP will perform assignment more efficiently.

Configuring DHCP Address Pool

DHCP Address assignment and DHCP parameters sent to the client should be defined in the DHCP address pool. If no DHCP address pool is configured, addresses cannot be assigned to the DHCP clients even though the DHCP server has been enabled. However, if the DHCP server has been enabled, the DHCP relay agent is always working regardless of the DHCP address pool.

You can give a meaningful name that can be memorized easily to the DHCP address pool. The name of address pool contains characters and digits. Ruijie product allows you to define multiple address pools. The IP address of the DHCP relay agent in the DHCP request packet is used to determine which address pool is used for address assignment.

- If a DHCP request packet contains no IP address of the relay agent, the address will be assigned according to the segment range of the interface receiving request packets. The logic of assignment is that, if an address pool of a large segment scope is configured, addresses can be assigned for the request packets received by the small segment interfaces within the large address pool segment scope. For example, if the large address pool configured is 192.168.0.0/16, it can be used to assign addresses to the DHCP requests arriving at the small segment interfaces 192.168.1.0/24, 192.168.2.0/24 and 192.168.4.0/24. If multiple address pools of small segments are configured, these pools can assign addresses to the request packets arriving at the large segment interface covering the small segments. For example, the two small address pools 192.168.1.0/24 and 192.168.3.0/24 can assign addresses to the DHCP requests arriving at the interface of 192.168.0.0/16. If the minimum match between the segment range of the interface receiving request packets and the segment range of the address pool is unsuccessful, the address assignment fails.
- If the DHCP request packet contains the IP address of the DHCP relay agent, the address that is in the same subnet or network as this address is assigned to the DHCP client. If no address pool is defined for this network segment, address assignment fails.

To configure a DHCP address pool, perform the following tasks as appropriate, of which the first three tasks are mandatory:

-

Configuring an Address Pool Name and Enter Configuration Mode

Use the following command to configure an address pool name and enter address pool configuration mode in global configuration mode.

Command	Function
Ruijie(config)# **ip dhcp pool** *dhcp-pool*	Configures an address pool name and enter address pool configuration mode.

To display address pool configuration mode, use the **Ruijie(dhcp-config)** command.

Configuring the Address Pool Network Number and Mask

To configure dynamic address binding, the subnet and its mask of the new address pool must be configured to provide the DHCP server with an address space that can be assigned to clients. Unless there is address exclusion configuration, the addresses in all the address pools can be assigned to clients. DHCP assigns addresses in the address pool in order. If an address exists in the DHCP binding table or is detected to have existed in the segment, the next address will be checked until a valid address is assigned.

Use the following command to configure the subnet and mask of an address pool in address pool configuration mode.

Command	Function
Ruijie(dhcp-config)# **network** *network-number mask*	Configures the network number and mask of a DHCP address pool

Configuring the Client Boot File

The client boot file is the boot mapping file required when the client boots up. The boot mapping file is usually the operating system to be downloaded to the DHCP client.

Use the following commands to configure the client boot file in address pool configuration mode.

Command	Function
Ruijie (dhcp-config)# **bootfile** *filename*	Configures the client boot file.

Configures the Default Gateway of the Client

The default gateway of the client is configured to function as the default gateway parameter that the server assigns to the client. The IP address of the default gateway must be in the same network as the IP address of the DHCP client.

Use the following commands to configure the default gateway of the client in address pool configuration mode.

Command	Function
Ruijie(dhcp-config)# **default-router** *address* [*address2…address8*]	Configures the default gateway.

Configuring the Default Gateway for the DHCP Client

When Ruijie devices assign DHCP addresses, default gateways to be assigned to clients can be either specified manually or assigned dynamically.

- If the default gateway of the address pool is specified manually, the gateway address manually specified is the default gateway of the client when a lease is obtained from the corresponding address pool.
- If no default gateway is configured, the default address type dynamically assigned is determined based on whether the VRRP address is configured to the interface that receives packets. If the VRRP address has been configured, the gateway is selected based on whether the request packets carry the field “relay”. If the request packet is forwarded by the relay, the segment of the field “relay” is used as the default gateway to issue; otherwise, the interface address selected by the longest match principle is the gateway to be issued.

Use the following command to configure the default gateway for the DHCP client in address pool configuration mode.

Command	Function
Ruijie(dhcp-config)# **default-router** *address* [*address2…address8*]	Configures the default gateway.

Configuring the Address Lease Period

The lease period of the addresses assigned to clients by the DHCP server is infinite for static address pools, and 1 day for other address pools, by default. The client should request to update when the lease period is going to expire. Otherwise, it cannot use this address when the lease period expires.

Use the following command to configure the address lease period in address pool configuration mode.

Command	Function
Ruijie(dhcp-config)# **lease** {*days* [*hours*] [*minutes*] | **infinite**}	Configures the address lease period.

Configure the startup server and boot file of the client. The client startup file is the boot image file that is used for client startup. Usually, after obtaining an IP address from the DHCP server, the DHCP client will download the boot image file from the startup server (usually the TFTP server) and initialize the device using the obtained configuration file. If no configuration file information is obtained, the device will be started up with the empty configuration.

Use the following commands to configure the download server and boot file of a client in address pool configuration mode.

Command	Function
Ruijie (dhcp-config)# **next-server** *address* [*address2…address8*]	Configures the download server address for client startup
Ruijie (dhcp-config)# **bootfile** *filename*	Configures the client boot file name

Configuring the Domain Name of the DHCP Client

The domain name of the DHCP client can be specified. In this way, the domain name suffix will be automatically added to the incomplete host name to form a complete host name when the DHCP client accesses the network resources using the host name.

Use the following command to configure the domain name of the DHCP client in address pool configuration mode.

Command	Function
Ruijie(dhcp-config)# **domain-name** *domain*	Configures the domain name.

Configuring the Domain Name Server

A domain name server (DNS) should be specified for domain name resolution when the DHCP client accesses the network resources using a host name.

Use the following command to configure a domain name server for the DHCP client in address pool configuration mode.

Command	Function
Ruijie(dhcp-config)# **dns-server** *address* [*address2…address8*]	Configures a DNS server.

Configuring the NetBIOS WINS Server

Windows Internet Naming Server (WINS) is a domain name resolution service from Microsoft used by the TCP/IP network to resolve a NetNBIOS name to an IP address. The WINS server runs in Windows NT. After started, the WINS server will receive a registration request from the WINS client. When the WINS client is being shut down, it will send a name release message to the WINS server to guarantee the consistency of available computers between the WINS database and the network.

Use the following command to configure a NetBIOS WINS server for the DHCP client in address pool configuration mode.

Command	Function
Ruijie(dhcp-config)# **netbios-name-server** *address* [*address2…address8*]	Configures a DNS server.

Configuring the NetBIOS Node Type for the DHCP Client

There are four types of NetBIOS nodes for Microsoft DHCP client: 1) Broadcast. The NetBIOS name is resolved in broadcast mode; 2) Peer-to-peer. The WINS server is asked directly to resolve the NetBIOS name; 3) Mixed. First, the name is resolved in broadcast mode, and then the WINS server is connected to resolve the name; 4) Hybrid. First the WINS server is asked directly to resolve the NetBIOS name. If there is no response, the NetBIOS name is resolved in broadcast mode.

By default, the Windows operation systems support the broadcast or hybrid type NetBIOS nodes. If no WINS server is configured, the node is of the broadcast type. If a WINS server is configured, the node is of the hybrid type.

Use the following command to configure the NetBIOS node type for the DHCP client in address pool configuration mode.

Command	Function
Ruijie(dhcp-config)# **netbios-node-type** *type*	Configures the NetBIOS node type.

Configuring the 6RD Parameters

The IPv6 rapid development (6RD) parameter includes the generic IPv4 prefix and suffix length, 6RD prefix length, 6RD prefix, and IPv4 address of the 6RD border relay (BR) for a given 6RD domain. If the DHCP client wants to create a 6RD tunnel, it can obtain the 6RD parameter via DHCP option 212.

Use the following command in DHCP address pool configuration mode to configure the 6RD parameter available for the DHCP client.

Command	Function
Ruijie(dhcp-config)# **option 6rd ipv4masklen** <*mask-length*> **ipv6prefixlen** <*prefix-length*> **ipv6prefix** <*ipv6-prefix*> **br-addr** <*ipv4-address*>	Configures the 6RD parameter.

Configuring the Network ID and Mask of the DHCP Address Pool

You must configure the subnet and its mask to provide the DHCP server with a client address assignment range when binding dynamic addresses. All address in this range can be allocated to clients if no excluded address is configured. DHCP assigns addresses in the range one by one. If an address exists in the DHCP binding table or is detected in the network segment, the DHCP proceeds to the next address until finding an effective allocable address.

Use the following commands to configure the subnet and its mask of the address pool in address configuration mode.

Command	Function
Ruijie(dhcp-config)# **network** *network-number mask*	Configures the network ID and mask of the DHCP address pool.

Caution

Address assignment is indexed by the physical address and ID of the client in the DHCP dynamic address pool of Ruijie products. In this case, the DHCP address pool cannot have two leases of the same client. Then, address assignment of the server becomes faulty and may result in failure to assign addresses if path redundancy occurs in the network topology between the server and the client (the client can connect to the server through both the direct path and the relay path).

Caution

To avoid the preceding problem, the network administrator can avoid path redundancy between the server and the client using methods such as adjusting the physical link or network link.

Configuring DHCP Address Pool to Allocate Address as per Option82

Generally, the DHCP relay agent will insert Option 82 to carry relevant information about the client during the process of packet forwarding (such as the VLAN to which the client belongs, slot number, port number or user's 1X class). Upon receipt of such packets, the DHCP server will allocate addresses according to the specific information about clients by analyzing Option 82 information. For example, Option 82 can be used to allocate an IP address range to clients belonging to a VLAN or user class. This feature can be used when required to allocate a specific range of IP addresses according to user's network allocation information (such as VLAN, slot number or port number) or user's priority.

Each DHCP address pool can allocate addresses using Option 82 information. Option 82 information will be matched and classified, and we can specify the allocable address range for the corresponding class. One DHCP address pool can be associated with multiple classes, and different network segment ranges can be specified for each class.

During the process of address allocation, we can first determine the allocable address pool according to the network segment to which the client belongs, and then further determine its CLASS according to Option 82 information, so as to allocate IP address from the network segment range corresponding to the CLASS. When a request packet matches multiple classes in the address pool, address will be assigned from the address ranges corresponding to these classes so that the classes are configured in the address pool. If the class has no allocable address, the network segment range for the next matching class will be used, and the like. Each class corresponds to one address range, and the addresses must be assigned from low to high. Multiple classes can be configured with the same address range. If the CLASS associated with the address pool is specified, but the segment range of the CLASS is not configured, the DHCP clients of this CLASS cannot be assigned addresses.

Use the following commands to configure the CLASS associated with address pool and the address range corresponding to the class in address pool configuration mode.

Command	Function
Ruijie(dhcp-config)# **class** *class-name*	Configures the name of associated class, and enter class configuration mode of address pool.

Command	Function
Ruijie(config-dhcp-pool-class)# **address range** *low-ip-address high-ip-address*	Configures the corresponding network segment range.

Note

1. When the global class cannot be found, it will be created automatically.
2. The associated class configured in the address pool may conflict with the static manual binding, and therefore they cannot be configured at the same time.
3. Up to five classes can be configured for each address pool

Configuring Class

Configuring Option82 Matching Information for the CLASS

The specific Option82 matching information corresponding to each CLASS can be configured after entering CLASS configuration mode in global mode. One CLASS can match multiple pieces of Option 82 information, and it is considered matched if the packet matches any information. If no matching information is configured for CLASS, then this CLASS can match any request packets carrying Option 82 information. The address can only be assigned from the corresponding address pool after the request packet matches a specific CLASS.

Use the following commands to configure global CLASS and the Option 82 information corresponding to the CLASS in global configuration mode.

Command	Function
Ruijie(config)# **ip dhcp class** *class-name*	Configures CLASS name and enters global CLASS configuration mode.
Ruijie(config-dhcp-class)# **relay agent information**	Enters Option 82 matching information configuration mode.
Ruijie(config-dhcp-class-relayinfo)# **relay-information hex** *aabb.ccdd.eeff…* [*****]	Configures specific Option 82 matching information: 1. The parameter *aabb.ccdd.eeff..* is a hexadecimal number. The asterisk (*) indicates imperfect matching mode. It is considered matched if the information before * is matched.

Note

The Global CLASS can have up to 20 matches.

Configuring Remark Information for the CLASS

Use the following commands to configure remark information to describe the meaning of CLASS in global configuration mode.

Command	Function
Ruijie(config)# **ip dhcp class** *class-name*	Configures CLASS name and enters CLASS configuration mode.
Ruijie(config-dhcp-class)#**remark** *used in #1 building*	Configures remark information.

Configuring whether to Use CLASS Allocation

Use the following command to configure address allocation using CLASS in global configuration mode.

Command	Function
Ruijie(config)# **ip dhcp use class**	Configures address allocation using CLASS.

Caution

This command is enabled by default. Execute NO command to disable address allocation using CLASS.

Configuring Binding Database Storage

Configuring to Periodically Save Binding Database into FLASH

To avoid the loss of binding database (lease information) on DHCP server due to power failure or reboot of the device, you can configure the delay time to write the database into FLASH. The time is **0** by default, namely the database will be written into FLASH at variable intervals.

Use the following command to periodically write the binding database into the FLASH in global configuration mode.

Command	Function
Ruijie(config)# [**no**] **ip dhcp database write-delay** [*time*]	Configures DHCP delay time to write into FLASH. *The range of the time* parameter is from 600—to 86400 in seconds. The default is 0.

Caution

Since frequent FLASH reading and writing will shorten the service life of FLASH, we shall pay attention to the delay time configured. Short delay will enable efficient storage of device information, while long delay can reduce the frequency of FLASH reading and writing, thus prolonging the service life.

Configuring to Manually Save the Binding Database into FLASH

To avoid the loss of DHCP binding database (lease information) due to power failure or reboot of the device, you can also manually write the existing binding database information into the FLASH as needed besides configuring the delay time for FLASH writing.

Use the following command to manually write the binding database into the FLASH in global configuration mode.

Command	Function
Ruijie(config)# **ip dhcp database write-to-flash**	Writes DHCP binding database information into the FLASH

Manual Address Binding

Address binding refers to mapping the IP address to the MAC address for the DHCP clients. You can bind addresses in two ways. 1) Manual binding: Configure the static IP address to MAC address mapping for the DHCP client on the DHCP server manually. Manual binding actually offers a special address pool; 2) Dynamic binding: Upon receiving a DHCP request from the DHCP client, the DHCP server dynamically assigns an IP address from the DHCP address pool to the DHCP client, and thus mapping the IP address to the MAC address for the DHCP client.

To define manual address binding, you first need to define a host address pool for each manual binding, and then define the IP address and hardware address (MAC address) or ID for the DHCP client. Generally, a client ID, instead of a MAC address, is defined for the Microsoft clients. The client ID contains media type and MAC address. For the codes of media types, see the section "Address Resolution Protocol Parameters" in RFC 1700. The code of Ethernet type is “01”.

Use the following commands to configure the manual address binding in address pool configuration mode.

Command	Function
Ruijie(config)# **ip dhcp pool** *name*	Defines the name of the DHCP address pool and enters DHCP configuration mode.
Ruijie(dhcp-config)# **host** *address* [*netmask*]	Defines an IP address for the DHCP client.
Ruijie(dhcp-config)# **hardware-address** *hardware-address type* *or:* Ruijie(dhcp-config)# **client-identifier** unique-identifier	Defines a hardware address for the DHCP client, such as aabb.bbbb.bb88. Defines an ID for the DHCP client, such as 01aa.bbbb.bbbb.88.
Ruijie(dhcp-config)# **client-name** *name*	(Optional) Defines the client name using standard characters of American Standard Code for Information Interchange (ASCII). Exclude the domain name in the client name. For example, if you define the host name as *mary*, do not define the client name as *mary.rg.com*.

Configuring Ping Times

By default, when trying to assign an IP address from the DHCP address pool to a DHCP client, the DHCP server will ping the IP address twice (one packet for each time). If there is no response, the DHCP server considers this address an idle address and assigns it to the DHCP client. If there is a response, the DHCP server considers that this address is in use and tries to assign another address to the DHCP client until an address is assigned successfully.

Use the following command to configure the number of Ping packets in global configuration mode.

Command	Function
Ruijie(config)# **ip dhcp ping** *packets number*	Configures the number of Ping packets before the DHCP server assigns an address. If it is set to **0**, the Ping operation is not performed. The default value is **2**.

Configuring Ping Packet Timeout

By default, the DHCP server considers an IP address nonexistent if no response is received within 500 milliseconds after pinging the IP address. You can adjust the Ping packet timeout.

Use the following command to configure the Ping packet timeout in global configuration mode.

Command	Function

Command	Function
Ruijie(config)# **ip dhcp ping timeout** *milliseconds*	Configures the Ping packet timeout for the DHCP server. The default value is 500ms.

Configuring the DHCP Client on the Ethernet Interface

☑ Ruijie products support the function of dynamically obtaining the IP address that is assigned by the DHCP server on an Ethernet interface.

Use the following command to configure the DHCP client on the Ethernet port, execute the following command in interface configuration mode.

Command	Function
Ruijie(config-if)# **ip address dhcp** [**6rd**]	Obtains an IP address through DHCP. The **6rd** parameter indicates that the device applies for the 6RD parameter configuration while requesting the IP address.

Configuring the DHCP Client in the PPP Encapsulation Link

Ruijie products support the function of dynamically obtaining the IP addresses that is assigned by the DHCP server on a PPP encapsulation interface.

Use the following command to configure the DHCP client, execute the following command in interface configuration mode.

Command	Function
Ruijie(config-if)# **ip address dhcp**	Obtains an IP address through DHCP.

Configuring the DHCP Client in the FR Encapsulation Link

☑ Ruijie products support obtaining the IP addresses dynamically assigned by the DHCP server on an FR encapsulation interface.

Use the following command to configure the DHCP client, execute the following command in interface configuration mode.

Command	Function
Ruijie(config-if)# **ip address dhcp**	Obtains an IP address through DHCP.

Configuring the DHCP Client in the HDLC Encapsulation Link

☑ Ruijie products support obtaining the IP address dynamically assigned by the DHCP server on an HDLC encapsulation interface.

Use the following command to configure the DHCP client, execute the following command in interface configuration mode.

Command	Function
Ruijie(config-if)# **ip address dhcp**	Obtains an IP address through DHCP.

☑ For some product of v10.1 and later versions, DHCP client supports obtaining the IP address assigned by the DHCP server in the point-to-point link of PPP, HDLC, and FR encapsulation.

Monitoring and Maintaining Information

Monitoring and Maintaining the DHCP Server

Three types of commands are available for monitoring and maintaining the DHCP server:

Clear commands, used to clear such information as DHCP address binding, address conflict and server statistics;
Debug commands, used to output necessary debugging information. Such commands are mainly used to diagnose and fix faults;
Show commands, used to show information about DHCP.

Ruijie products provide three clear commands. Use the following commands to clear information in command execution mode.

Command	Function
Ruijie# **clear ip dhcp binding** { *address* | *****}	Clears the DHCP address binding information.
Ruijie# **clear ip dhcp conflict** { *address* | *}	Clears the DHCP address conflict information.
Ruijie# **clear ip dhcp server statistics**	Clears the DHCP server statistics.

Use the following command to debug the DHCP server in command execution mode.

Command	Function
Ruijie# **debug ip dhcp server** [**events | packet**]	Debugs the DHCP server.

Use the following commands to show the working status of the DHCP server in command execution mode.

Command	Function
Ruijie# **show ip dhcp binding** [*address*]	Shows the DHCP address binding information.
Ruijie# **show ip dhcp conflict**	Shows the DHCP address conflict information.
Ruijie# **show ip dhcp server statistics**	Shows the DHCP server statistics.

Monitoring and Maintaining the DHCP Client

There are two types of commands for monitoring and maintaining the DHCP client. The following operations can be performed on the DHCP client:

Debug commands, used to output necessary debugging information. Such commands are mainly used to diagnose and clear faults.
Show commands, used to show information about DHCP.

Use the following command to debug the DHCP client, execute the following command in command execution mode.

Command	Function
Ruijie# **debug ip dhcp client**	Debugs the DHCP client.

Use the following command to show information about the lease of the DHCP client in command execution mode.

Command	Function
show dhcp lease	Shows the information about DHCP lease.

Example of Configuring the Address Pool to Support Option82

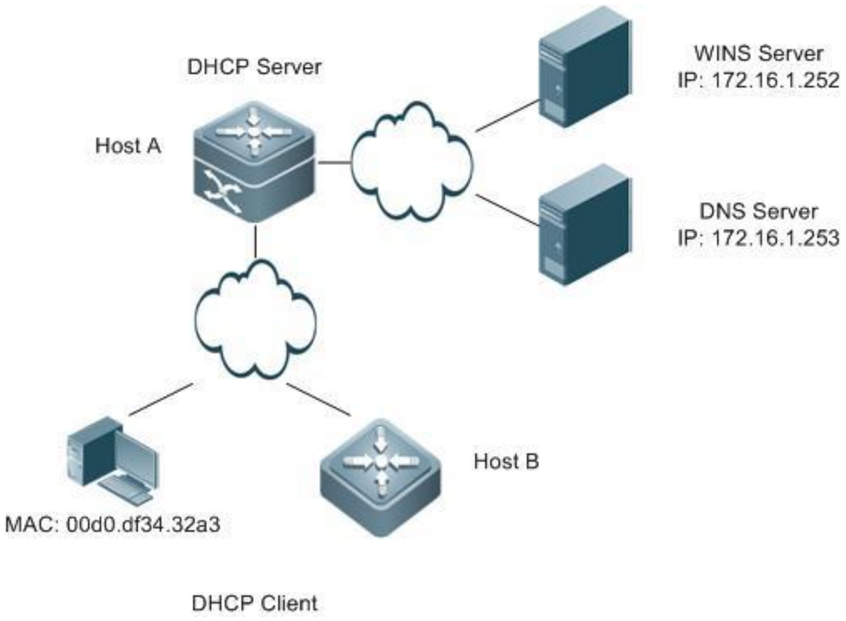
In the following example, an address pool of "net82" is defined; the address pool is in the network segment of 172.16.1.0/24, and the associated classes include class1, class2, class3 and class4. Class1 will allocate addresses from the range of 172.16.1.1-172.16.1.8; class2 will allocate addresses from the range of 172.16.1.9-172.16.1.18; class3 will allocate addresses from the range of 172.16.1.19-172.16.1.28; class4 has no defined address range, and will allocate addresses from the range of entire network segment. Configure class1 to match Option 82 information of 0100002120, class2 to match 0106020145, class3 to match 06020506*, and class4 to match any information.

```
!
ip dhcp class class1
 relay agent information
   relay-information hex 0100002120
!
ip dhcp class class2
 relay agent information
   relay-information hex 0106020145
!
ip dhcp class class3
 relay agent information
   relay-information hex 06020506*
!
ip dhcp class class4
!
ip dhcp pool net82
network 172.16.1.0 255.255.255.0
class class1
address range 172.16.1.1 172.16.1.8
class class2
address range 172.16.1.9 172.16.1.18
class class3
address range 172.16.1.19 172.16.1.28
class class4
```

Typical DHCP Configuration Examples

Topological Diagram

Fig 1-6 Diagram of DHCP example



Application Requirements

- Host A can serve as the DHCP server to assign dynamic IP addresses to some client users. The network segment for IP address assignment is 172.16.1.0/24; the default gateway is 172.16.1.254; the domain name is ruijie.com; the domain name server is 172.16.1.253; the WINS server is 172.16.1.252; the NetBIOS node type is compound; and the address lease period is 1 day. Except the addresses from 172.16.1.2 to 172.16.1.100 in the address segment, all the other addresses can be assigned.
- Host A assigns fixed IP addresses to some client users. The IP address assigned to the fit AP (DHCP client) with the MAC address 00d0.df34.32a3 is 172.16.1.101; the mask is 255.255.255.0; the domain name is admin; the default gateway is 172.16.1.254; the domain name server is 172.16.1.253; the WINS server is 172.16.1.252; and the NetBIOS node type is compound.
- Host B configures the DHCO automatically assigned address to the device interface FastEthernet 0/0.

Configuration Tips

- Enable the function of DHCP server on Host A and create an address pool to dynamically assign IP addresses. And create another address pool to manually bind IP addresses. Specify the address of the domain name server in the corresponding address pool (in this example, the addresses of the DNS server and WINS server) and the domain name of the client.
- Enable the function of the DHCP client on Host B to obtain the IP address automatically.

Configuration Steps

Step 1: Create a new DHCP address pool and configure dynamic IP address allocation on the Host A.

! Configure the name of address pool as "dynamic" and enter DHCP configuration mode.

```
HostA# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
HostA(config)# ip dhcp pool dynamic
```

! In DHCP configuration mode, configure an IP address network allocable to clients, configure the default gateway of this network segment, and set the lease period to 1 day.

```
HostA(dhcp-config)# network 172.16.1.0 255.255.255.0
HostA(dhcp-config)# default-router 172.16.1.254
HostA(dhcp-config)# lease 1
```

Step 2: Specify the DNS Server of "dynamic" address pool and configure the domain name of client.

! Assuming that the IP address of DNS Server is 172.16.1.253, configure Domain Name Server in the address pool and configure the domain name of client as ruijie.com.

```
HostA(dhcp-config)# dns-server 172.16.1.253
HostA(dhcp-config)# domain-name ruijie.com
```

Step 3: Specify the WINS Server of "dynamic" address pool and configure the NetBIOS node type of client.

! Assuming that the IP address of WIN Server is 172.16.1.252, configure NetBIOS WINS server in the address pool and configure the NetBIOS node type as Hybrid.

```
HostA(dhcp-config)# netbios-name-server 172.16.1.252
HostA(dhcp-config)# netbios-node-type h-node
```

Step 4: Configure excluded addresses in global mode.

! As shown in the preceding information, IP addresses of 172.16.1.254, 172.16.1.253 and 172.16.1.252 have been assigned to the gateway, the DNS server and the WINS server, and the addresses from 172.16.1.2 to 172.16.1.100 are excluded addresses. The excluded addresses won't be assigned to clients.

```
HostA(dhcp-config)# exit
HostA(config)# ip dhcp excluded-address 172.16.1.252 172.16.1.254
HostA(config)# ip dhcp excluded-address 172.16.1.2 172.16.1.100
```

Step 5: Create another address pool and manually bind the IP address.

! Configure the name of address pool as "static" and enter DHCP configuration mode.

```
HostA(config)# ip dhcp pool static
```

! Manually bind the IP address of 172.16.1.101/24 to the MAC address of 00d0.df34.32a3, with client name being "admin". Note: The identifier for identifying the client shall indicate the network media type ("01" for Ethernet), namely the identifier of the client corresponding to the manually bound MAC address shall be 00d0.df34.32a3.14.

```
HostA(dhcp-config)# host 172.16.1.101 255.255.255.0
HostA(dhcp-config)# client-identifier 00d0.df34.32a3.14
HostA(dhcp-config)# client-name admin
```

Step 6: Specify the gateway address corresponding to the "static" address pool.

! Configure gateway address as 172.16.1.254.

```
HostA(dhcp-config)# default-router 172.16.1.254
```

Step 7: Specify the DNS Server of "static" address pool and configure the domain name of client.

! Assuming that the IP address of the DNS server is 172.16.1.253,configure the DNS in the address pool and configure the domain name of client as ruijie.com.

```
HostA(dhcp-config)# dns-server 172.16.1.253
```

HostA(dhcp-config)# domain-name ruijie.com Step 8: Specify the WINS Server of "static" address pool and configure the NetBIOS node type of client.

! Assuming that the IP address of WIN Server is 172.16.1.252, configure NetBIOS WINS server in the address pool and configure the NetBIOS node type as Hybrid.

```
HostA(dhcp-config)# netbios-name-server 172.16.1.252
HostA(dhcp-config)# netbios-node-type h-node
```

HostA(dhcp-config)# exit Step 9: Enable the DHCP server on HOST A.

```
HostA(dhcp-config)# exit
```

HostA(config)# service dhcp Step 10: Enable the DHCP client on Host B.

！The following example shows how to enable the DHCP client, assuming that the client uses a layer 3 interface by default.

```
HostB(config)# interface fastEthernet 0/1
HostB(config-if-fastEthernet 0/1)# ip address dhcp
```

Verification

Step 1: View the configuration information on Host A.

```
HostA# show running-config
!
service dhcp
!
ip dhcp excluded-address 172.16.1.252 172.16.1.254
ip dhcp excluded-address 172.16.1.2 172.16.1.100
!
!
ip dhcp pool dynamic
 netbios-node-type n-node
 netbios-name-server 172.16.1.252
 domain-name ruijie.com
 lease 1 0 0
 network 172.16.1.0 255.255.255.0
 dns-server 172.16.1.253
 default-router 172.16.1.254
!
ip dhcp pool static
 client-name admin
 client-identifier 00d0.df34.32a3.14
 host 172.16.1.101 255.255.255.0
 netbios-node-type n-node
 netbios-name-server 172.16.1.252
 domain-name ruijie.com
 dns-server 172.16.1.253
 default-router 172.16.1.254
!
```

Step 2: View the configuration information on Host B.

```
HostB# show running-config
!
interface fastEthernet 0/1
 // Note: For switches, the no switchport command is also required, and the interface is configured as a layer 3 interface.
ip address dhcp
```

Step 3: Connect a PC with the MAC address 0013.2049.9014, and view the IP address information assigned by the DHCP server on the Host A.

```
Ruijie#show ip dhcp binding
IP address    Client-Identifier/    Lease expiration       Type
              Hardware address
172.16.1.101  00d0.df34.32a3.14     IDLE                   Manual 172.16.1.102
0100.e04c.70b7.e2 000 days 23 hours 48 mins Automatic
```

Configuring DHCP Relay

Overview

Understanding DHCP

The Dynamic Host Configuration Protocol (DHCP) is widely used to dynamically allocate reusable network resources such as IP addresses.

The DHCP client sends the DHCP DISCOVER broadcast packet to the DHCP server. After receiving the DHCP DISCOVER broadcast packet, the DHCP server allocates resources such as IP addresses to the DHCP client according to the appropriate policy, and sends the DHCP OFFER packet. After receiving the DHCP OFFER packet, the DHCP client checks if the resources are available. If yes, the DHCP client sends the DHCP REQUEST packet. If no, the DHCP client sends the DHCP DISCOVER packet. After receiving the DHCP REQUEST packet, the DHCP server checks if the IP addresses (or other limited resources) can be allocated. If yes, the DHCP server sends the DHCP ACK packet. If no, the DHCP server sends the DHCP NAK packet. After receiving the DHCP ACK packet, the DHCP client starts to use the resources allocated by the DHCP server. Upon receiving the DHCP NAK packet, the DHCP client may re-send the DHCP DISCOVER packet to request another IP address.

Understanding DHCP Relay

The destination IP address of DHCP REQUEST packet is 255.255.255.255. Such packets are only forwarded inside a subnet. To allocate IP addresses dynamically across network segments, the DHCP relay agent comes into being. It encapsulates the received DHCP REQUEST packet into unicast IP packets and forwards it to the DHCP server. Meanwhile, it forwards the received DHCP response packet to the DHCP client. In this way, the DHCP Relay Agent works as a transit station responsible for communicating with the DHCP clients and the DHCP server on different network segments. In this case, one DHCP server in a LAN can implement the dynamic IP management for all network segments, that is, a dynamic DHCP IP management in Client - Relay Agent - Server mode, as shown in Figure 1.

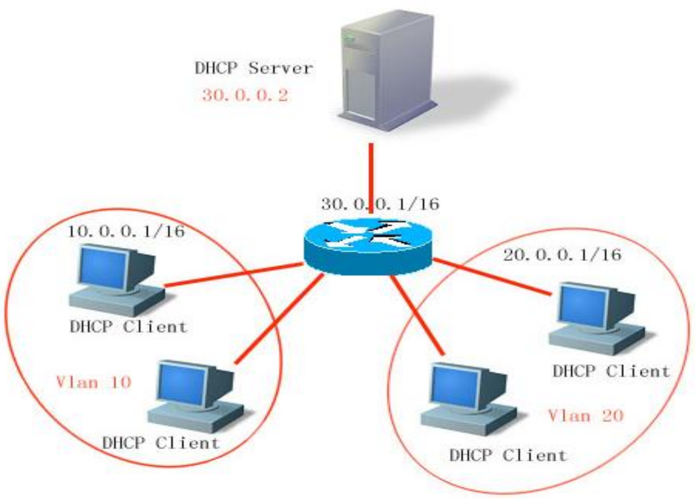


Figure 1

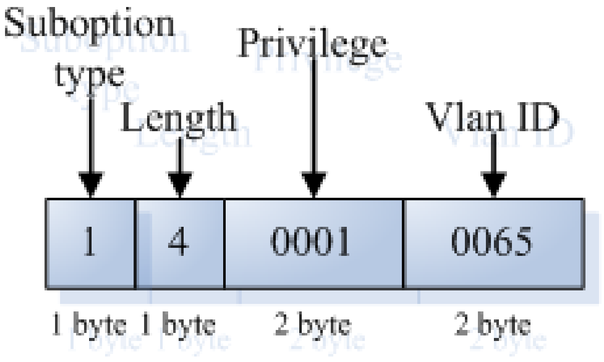
VLAN 10 and VLAN 20 correspond with the 10.0.0.1/16 and 20.0.0.1/16 networks respectively, while the DHCP server is located on the 30.0.0.1/16 network. To have a dynamic IP management on the 10.0.0.1/16 and 20.0.0.1/16 networks through the DHCP server at 30.0.0.2, just enable the DHCP Relay Agent on the device that functions as the gateway, and specify the IP address of the DHCP server to 30.0.0.2.

Understanding DHCP Relay Agent Information (option82)

As specified in RFC3046, when a relay device performs DHCP relay, an option can be added to identify the network information of the DHCP client, so that the DHCP server can assign users with IP addresses of different privileges. RFC3046 specifies that the option is numbered 82, so it is also called option82. This option can be divided into several sub-options. Currently, the sub-options frequently used are Circuit ID and Remote ID. Ruijie Networks provides threetwo schemes for relay agent information, which are described as follows.

relay agent information option dot1x: This requires the combination of 802.1x authentication and Ruijie RG-SAM.

DHCP relay forms the Circuit ID sub-option by combining the IP priority assigned to RG-SAM in 802.1x authentication with VID of the DHCP client. Figure 2 shows the option format.



relay agent information option82: This can be used without running other protocol modules. During DHCP relay, the device forms option82 information according to the port that receives the DHCP request message and the physical IP address of the device, and uploads the option82 information to the DHCP server.

Figure 3 and Figure 4 show the option formats.

Agent Circuit ID

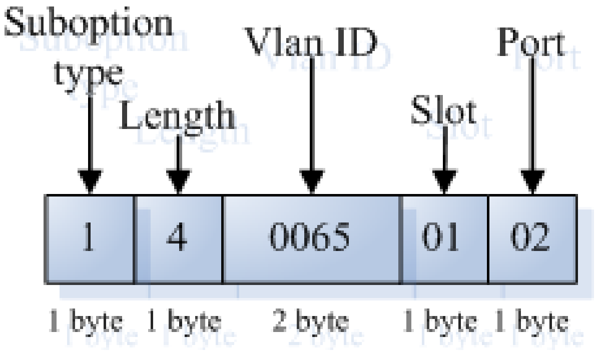


Figure 3

Agent Remote ID

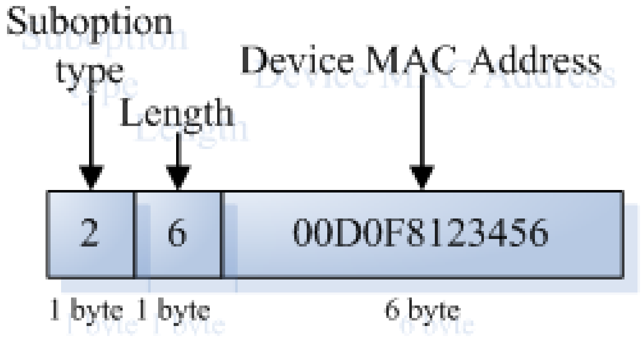


Figure 4

Understanding the DHCP Relay Check Server-id Function

This section describes the DHCP relay check server-id function. When DHCP is used, multiple DHCP servers are configured for a network for backup, so that the network will continue to work even if a server fails. During the four interaction processes of DHCP acquisition, a DHCP server has been selected when the DHCP client sends the DHCP request message. The DHCP request message includes the optional server-id. In particular application circumstances, you need to enable this option for relay to reduce loads on the network server. In this way, the DHCP request message is only sent to the specified DHCP server.

Configuring DHCP

Configuring the DHCP Relay Agent

Use the following commands to configure the DHCP relay agent in global configuration mode.

Command	Function
Ruijie (config)# **service dhcp**	Enables the DHCP agent.
Ruijie(config)# **no service dhcp**	Disables the DHCP agent.

Configuring the IP Address of the DHCP Server

After you have configured the IP address of the DHCP server, the DHCP request message received by the device will be forwarded to the DHCP server. Meanwhile, the DHCP response message from the DHCP server will be forwarded to the DHCP client.

The IP address of the DHCP server can either be configured globally or on the layer 3 interface. Up to 20 IP addresses can be configured for the DHCP server in each mode. When the DHCP request message is received from an interface, the DHCP server list on the interface is at first. If no DHCP server list is configured on the interface, the DHCP server list globally configured will be used.

DHCP supports vrf-based relay by adding the *vrf* parameter to the IP address of the DHCP server.

Use the following commands to configure the IP address of the DHCP server in global configuration mode.

Command	Function
Ruijie(config)# **ip helper-address** [**vrf** { *vrf-name*} **|** **global**] *A.B.C.D*	Adds the IP address of the DHCP server globally. The VPN or global space of the specified server can be displayed.
Ruijie(config-if)# **ip helper-address** [**vrf** {*vrf-name*} **|** **global**] *A.B.C.D*	Adds the IP address of the DHCP server on the interface. This command must be set on the layer 3 interface. The VPN or global space of the specified server can be displayed. The server is of the same VPN or global space as the current interface by default.
Ruijie(config)# **no IP helper-address** [**vrf** { *vrf-name*} **|** **global**] *A.B.C.D*	Deletes the globally configured IP address of the DHCP server.
Ruijie(config-if)# **no IP helper-address** [**vrf** { *vrf-name* } **| global**] *A.B.C.D*	Deletes the IP address of the DHCP server configured on the interface.

Configuring DHCP option dot1x

The section "Understanding the DHCP Relay Agent Information" shows that you can configure the **ip dhcp relay information option dot1x** command to enable the **option dot1x** function of DHCP relay when you need to assign the IP addresses with different privileges to the users of different privileges. When this function is enabled, the device will work with 802.1x to add corresponding option information to the DHCP server when it relays. This function should be used with the dot1x function.

Use the following commands to configure DHCP option dot1x in global configuration mode.

Command	Function
Ruijie(config)# **ip dhcp relay information option dot1x**	Enables the DHCP option dot1x function.
Ruijie(config)# **no ip dhcp relay information option dot1x**	Disables the DHCP option dot1x function.

Configuring DHCP option dot1x access-group

In the option dot1x application scheme, the device needs to restrict the unauthorized IP address or the IP address with low privilege to access certain IP addresses, and restrict the access between users with low privileges. To do so, configure the **ip dhcp relay information option dot1x access-group** *acl-name* command. The Access Control List (ACL) defined by *acl-name* must be configured in advance. It is used to filter some contents and prohibit unauthorized users from accessing each other. In addition, the ACL associated here is applied to all the ports on the device. This ACL has no default Access Control Entry (ACE) and does not conflict with ACLs associated with other interfaces. For example:

Assign a type of IP addresses for all the unauthorized users, namely 192.168.3.2-192.168.3.254, 192.168.4.2-192.168.4.254, and 192.168.5.2-192.168.5.254. Do not assign gateway addresses 192.168.3.1, 192.168.4.1, and 192.168.5.1 to users. In this way, an unauthorized user uses one of the 192.168.3.x-5.x addresses to access the web portal for downloading the client software. The device should be configured as follows:

```
Ruijie# configure terminal
Ruijie(config)# ip access-list extended DenyAccessEachOtherOfUnauthrize
Ruijie(config-ext-nacl)# permit ip any host 192.168.3.1
```

//Permit the packet to be transmitted to the gateway.

```
Ruijie(config-ext-nacl)# permit ip any host 192.168.4.1
Ruijie(config-ext-nacl)# permit ip any host 192.168.5.1
Ruijie(config-ext-nacl)# permit ip host 192.168.3.1 any
```

//Permit the packet communication with the source IP address being the gateway.

```
Ruijie(config-ext-nacl)# permit ip host 192.168.4.1 any
Ruijie(config-ext-nacl)# permit ip host 192.168.5.1 any
Ruijie(config-ext-nacl)# deny ip 192.168.3.0 0.0.0.255 192.168.3.0 0.0.0.255
```

//Prohibit mutual accesses of unauthorized users.

```
Ruijie(config-ext-nacl)# deny ip 192.168.3.0 0.0.0.255 192.168.4.0  0.0.0.255
Ruijie(config-ext-nacl)# deny ip 192.168.3.0 0.0.0.255 192.168.5.0  0.0.0.255
Ruijie(config-ext-nacl)# deny ip 192.168.4.0 0.0.0.255 192.168.4.0  0.0.0.255
Ruijie(config-ext-nacl)# deny ip 192.168.4.0 0.0.0.255 192.168.5.0  0.0.0.255
Ruijie(config-ext-nacl)# deny ip 192.168.5.0 0.0.0.255 192.168.5.0  0.0.0.255
Ruijie(config-ext-nacl)# deny ip 192.168.5.0 0.0.0.255 192.168.3.0  0.0.0.255
Ruijie(config-ext-nacl)# deny ip 192.168.5.0 0.0.0.255 192.168.4.0  0.0.0.255
Ruijie(config-ext-nacl)# exit
```

Then, apply the command to the global interfaces using the **ip dhcp relay information option dot1x access-group** *DenyAccessEachOtherOfUnauthrize* command.

Use the following commands to configure **DHCP option dot1x access-group** in global configuration mode.

Command	Function
Ruijie(config)# **ip dhcp relay information option dot1x access-group** *acl-name*	Enables DHCP option dot1x acl.
Ruijie(config)# **no ip dhcp relay information option dot1x access-group** *acl-name*	Disables DHCP option dot1x acl.

Configuring DHCP option82

When the **ip dhcp relay information option82** command is configured, the device, as DHCP relay, adds option information in the DHCP request packet to the DHCP server during forwarding the request packet.

Use the following commands to configure DHCP option82 in global configuration mode:

Command	Function
Ruijie(config)# ip dhcp relay information option82	Enables the DHCP option82 function.
Ruijie(config)# no ip dhcp relay information option82	Disables the DHCP option82 function.

Configuring DHCP relay check server-id

After the **ip dhcp relay check** *server-id* command is configured, the device resolves the *dhcp server-id* option upon receiving DHCP relay. If this option is set, the DHCP request message is sent to this server only.

Use the following commands to configure **DHCP relay check** *server-id* function in global configuration mode.

Command	Function
Ruijie(config)# **ip dhcp relay check** *server-id*	Enables the DHCP relay check server-di function.
Ruijie(config)# **no ip dhcp relay check** *server-id*	Disables the DHCP relay check server-id function.

Configuring DHCP Relay Suppression

After the **ip dhcp relay suppression** command is configured, the port will not relay the DHCP request broadcast packet by transforming it into the unicast form. However, it will not suppress the normal forwarding of broadcast packets received.

Use the following commands to configure DHCP relay suppression in interface configuration mode.

Command	Function
Ruijie(config-if)# **ip dhcp relay Suppression**	Enables the DHCP relay suppression function.
Ruijie(config-if)# **no ip dhcp relay Suppression**	Disables the DHCP relay suppression function.

DHCP Relay Configuration Example

The following commands enable the DHCP relay function and add two groups of IP addresses of the DHCP server:

```
Ruijie# configure  terminal
Ruijie(config)# service dhcp       //Enable the dhcp relay function
Ruijie(config)# service dhcp             //Enable the DHCP option vpn function.
Ruijie(config)# ip helper-address 192.18.100.1
Ruijie(config)# ip helper-address 192.18.100.2
  //Add the IP address of the server at the interface.
Ruijie(config)# interface gigabitEthernet 0/3
Ruijie(config-if-gigabitEthernet 0/3)# ip helper-address 192.18.200.1
Ruijie(config-if-gigabitEthernet 0/3)# ip helper-address 192.18.200.2
Ruijie(config-if-gigabitEthernet 0/3)# end
```

Other Precausions on DHCP Relay Configuration

For layer 2 network devices, you must enable at least one of the option dot1x, dynamic address binding, and option82 functions when the cross network segment management vlan relay function is required. Otherwise, only the relay function of management VLAN can be enabled for the layer 2 device.

Precautions on DHCP option dot1x Configuration

This command works only when the configuration related to AAA/802.1x is correct.

When this scheme is adopted, the IP authorization of the DHCP mode of 802.1x should be enabled.

This command cannot be used together with the **dhcp option82** command because they are conflicted.

When the IP authorization of the DHCP mode of 802.1x is enabled, the MAC address and the IP address will also be bound. Therefore, IP authorization and DHCP dynamic binding function cannot be enabled at the same time.

Precautions on DHCP option82 Configuration

The DHCP option82 function and the **dhcp option dot1x** function cannot be used at the same time because they are conflicted.

Showing the DHCP Configuration

Use the **show running-config** command to show the DHCP configuration in privileged mode.

```
Ruijie# show running-config
Building configuration...
Current configuration : 1464 bytes
version RGOS 10.1.00(1), Release(11758)(Fri Mar 30 12:53:11 CST 2007 -nprd
hostname Ruijie
vlan 1
ip helper-address 192.18.100.1
ip helper-address 192.18.100.2
ip dhcp relay information option dot1x
interface GigabitEthernet 0/1
interface GigabitEthernet 0/2
interface GigabitEthernet 0/3
no switchport
ip helper-address 192.168.200.1
ip helper-address 192.168.200.2
interface VLAN 1
ip address 192.168.193.91 255.255.255.0
line con 0
exec-timeout 0 0
line vty 0
exec-timeout 0 0
login
password 7 0137
line vty 1 2
login
password 7 0137
line vty 3 4
```

```
login
end
```

Typical DHCP Relay Configuration Examples (for Switches)

Topological Diagram

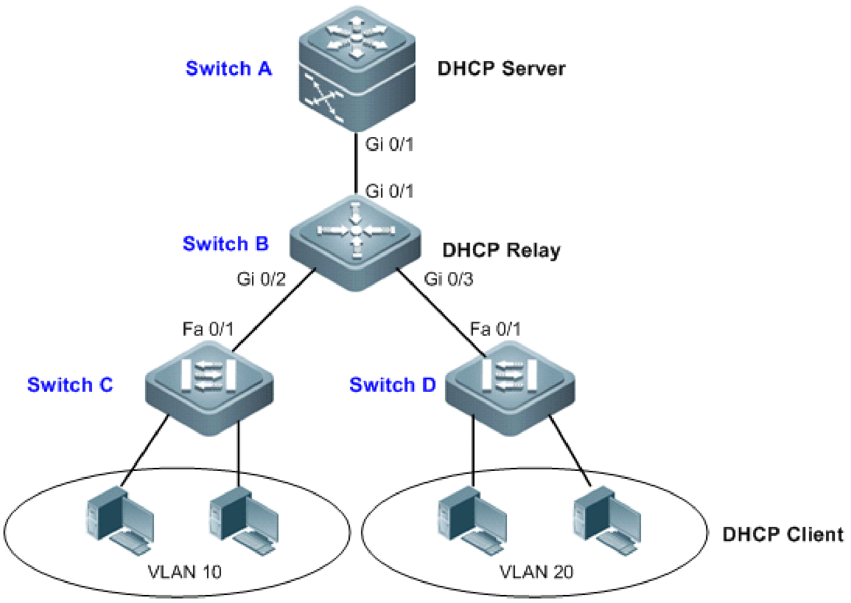


Diagram for DHCP relay configuration

Application Requirements

As shown in the preceding diagram, Switch C and Switch D are access devices connecting with PC users belonging to VLAN 10 and VLAN 20. Switch B is the gateway device, while Switch A is the core routing device. The following requirements must be met:

- Switch A can serve as DHCP server allocating dynamic IP addresses to VLAN users.
- The users connecting to Switch C and Switch D can acquire dynamic IP addresses across the network segment.

Configuration Tips

- **Configuring the DHCP server:** On Switch A, create DHCP address pools for users from VLAN 10 and VLAN 20 respectively, and enable the DHCP server (relevant configurations of the DHCP server can be found in the section "DHCP Configuration").
- **Configuring DHCP Relay:** On Switch B, configure the address of the DHCP server (configure the address of the DHCP server as 10.1.1.2/24) and enable the DHCP server.

Note

On Switch C and Switch D, configure the VLAN to which the corresponding ports belong, and the access PC can dynamically acquire IP address once connected.

Configuration Steps

Configure the DHCP server.

! In global mode, create a DHCP address pool named "vlan10" on Switch A, with corresponding IP network segment being 192.168.1.0/24 and the address of network gateway being 192.168.1.1.

```
SwitchA(config)#ip dhcp pool vlan10
SwitchA(dhcp-config)#network 192.168.1.0 255.255.255.0
SwitchA(dhcp-config)#default-router 192.168.1.1
SwitchA(dhcp-config)#exit
```

! Create an address pool named "vlan20", with IP network segment being 192.168.2.0/24 and gateway address being 192.168.2.1.

```
SwitchA(config)#ip dhcp pool vlan20
SwitchA(dhcp-config)#network 192.168.2.0 255.255.255.0
SwitchA(dhcp-config)#default-router 192.168.2.1
SwitchA(dhcp-config)#exit
```

! In global configuration mode, configure 192.168.1.1 and 192.168.2.1 as the excluded addresses, so as to avoid the conflict between allocated IP address and gateway address.

```
SwitchA(config)#ip dhcp excluded-address 192.168.1.1
SwitchA(config)#ip dhcp excluded-address 192.168.2.1
```

! Enable the DHCP server.

```
SwitchA(config)#service dhcp
```

Step 2: Configure layer-3 communication between Switch A and Switch B.

! On Switch A, configure port Gi 0/1 as the Route Port, with the corresponding IP address being 10.1.1.2/24.

```
SwitchA(config)#interface gigabitEthernet 0/1
SwitchA(config-if-GigabitEthernet 0/1)#no switchport
SwitchA(config-if-GigabitEthernet 0/1)#ip address 10.1.1.2 255.255.255.0
SwitchA(config-if-GigabitEthernet 0/1)#exit
```

! On Switch B, configure port Gi 0/1 as the Route Port, with the corresponding IP address being 10.1.1.3/24.

```
SwitchB(config)#interface gigabitEthernet 0/1
SwitchB(config-if)#no switchport
SwitchB(config-if)#ip address 10.1.1.3 255.255.255.0
SwitchB(config-if)#exit
```

! Configure default route on Switch A

```
SwitchA(config)#ip route 0.0.0.0 0.0.0.0 10.1.1.3
```

Step 3: Configure the gateway for access users.

! On Switch B, configure the Switch Virtual Interface (SVI) of VLAN 10 to 192.168.1.1/24.

```
SwitchB(config)#vlan 10
SwitchB(config-vlan)#exit
SwitchB(config)#interface vlan 10
SwitchB(config-if)#ip address 192.168.1.1 255.255.255.0
SwitchB(config-if)#exit
```

! Configure the SVI of VLAN 20 to 192.168.2.1/24.

```
SwitchB(config)#vlan 20
SwitchB(config-vlan)#exit
SwitchB(config)#interface vlan 20
SwitchB(config-if)#ip address 192.168.2.1 255.255.255.0
SwitchB(config-if)#exit
```

Step 4: Configure DHCP Relay.

! On Switch B, globally configure the address of DHCP server as 10.1.1.2 and enable the DHCP server.

```
SwitchB(config)#ip helper-address 10.1.1.2
SwitchB(config)#service dhcp
```

Step 5: Configure layer-2 communication between Switch B and Switch C/D.

! On Switch B, configure ports Gi 0/2 and Gi 0/3 as the Trunk Port.

```
SwitchB(config)#interface range gigabitEthernet 0/2-3
SwitchB(config-if-range)#switchport mode trunk
```

! Configure port Fa 0/1 of Switch C and Switch D as the Trunk Port.

Verification

Step 1: Show configurations of devices.

! Configurations of Switch A

```
SwitchA#show running-config
!
service dhcp
!
ip dhcp excluded-address 192.168.1.1
ip dhcp excluded-address 192.168.2.1
!
ip dhcp pool vlan10
 network 192.168.1.0 255.255.255.0
 default-router 192.168.1.1
!
ip dhcp pool vlan20
 network 192.168.2.0 255.255.255.0
 default-router 192.168.2.1
!
```

```
interface GigabitEthernet 0/1
 no switchport
 no ip proxy-arp
 ip address 10.1.1.2 255.255.255.0
!
ip route 0.0.0.0 0.0.0.0 10.1.1.3
!
! Configurations of Switch B
SwitchB#show running-config
!
vlan 10
!
vlan 20
!
service dhcp
ip helper-address 10.1.1.2
!
interface GigabitEthernet 0/1
 no switchport
 no ip proxy-arp
 ip address 10.1.1.3 255.255.255.0
!
interface GigabitEthernet 0/2
 switchport mode trunk
!
interface GigabitEthernet 0/3
 switchport mode trunk
!
interface VLAN 10
 no ip proxy-arp
 ip address 192.168.1.1 255.255.255.0
!
interface VLAN 20
 no ip proxy-arp
 ip address 192.168.2.1 255.255.255.0
!
```

Step 2: Connect two PCs with the ports belonging to VLAN 10 and VLAN 20 and verify dynamic IP address allocation.

```
SwitchA#show ip dhcp binding
IP address Client-Identifier/ Lease expiration  Type Hardware address
192.168.1.2 0100.1320.4990.14 000 days 23 hours 59 mins  Automatic
192.168.2.2 0100.e04c.70b7.e2 000 days 23 hours 59 mins  Automatic
```

Typical DHCP Relay Configuration Examples (for Routers)

Topological Diagram

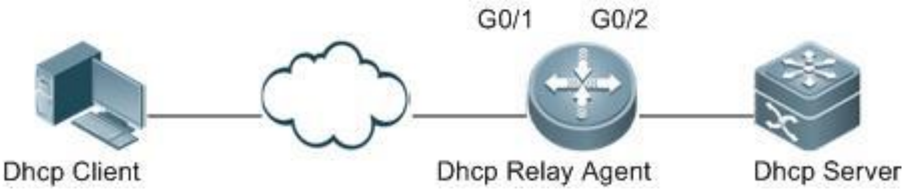


Diagram for DHCP Relay configuration

Application Requirements

As shown in the preceding diagram, obtaining the IP address and surfing the Internet by the user in different network segment shall be implemented when the DHCP Relay function is enabled.

Configuration Tips

- Enable the function of acquiring IP addresses through DHCP.
- Enable the DHCP Relay function on the DHCP Relay Agent.
- Configure the DHCP server.

Configuration Steps

Enable DHCP to acquire IP addresses.

Configure DHCP Relay:

# Enable the DHCP Relay Agent.

```
Ruijie(config)# server dhcp
```

# Add an IP address of DHCP server globally.

```
Ruijie(config)# ip helper-address 172.2.2.1
```

# Configure an IP address of the port connecting the user device.

```
Ruijie(config)# interface gigabitEthernet 0/1
Ruijie(config-if)# ip address 192.1.1.1 255.255.255.0
```

# Configure an IP address for the port connecting the user device.

```
Ruijie(config)# interface gigabitEthernet 0/1
Ruijie(config-if)# ip address 192.1.1.1 255.255.255.0
```

# Configure an IP address for the port connecting the Server device.

```
Ruijie(config)# interface gigabitEthernet 0/2
Ruijie(config-if-gigabitEthernet 0/2)# ip address 172.2.2.2 255.255.255.0
```

Configure the DHCP server.

Verification

Verify configurations of the DHCP Relay Agent device.

# Log in to the DHCP Relay Agent device, and use the **show running-config** command in privileged mode to show the DHCP Relay configuration.

```
Ruijie# show running-config
service dhcp
ip helper-address 172.2.2.1
!
interface GigabitEthernet 0/1
ip address 192.1.1.1 255.255.255.0
!
interface GigabitEthernet 0/2
ip address 172.2.2.2 255.255.255.0
!
```

Typical Option dot1x Configuration Example (for Switches)

Topological Diagram

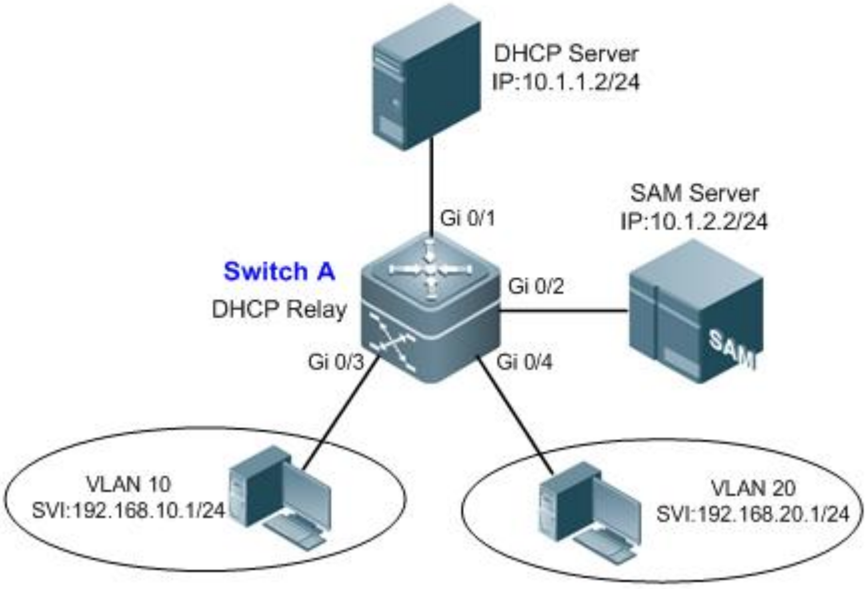


Diagram for DHCP Option Dot1x

Application Requirements

- Switch A is a layer 3 device allowing route communication cross different network segments.

- Access users belonging to different VLANs access Internet after Dot1x authentication, and SAM Server assigns different access privileges to different users.
- The DHCP server can allocate IP addresses to users according to the privilege of an authenticated user.

Configuration Tips

- **Configure basic DHCP Relay:** On Switch A, configure the address of the DHCP server (10.1.1.2/24) and enable the DHCP server. After configuration, the user can acquire dynamic IP address across the network segment.
- **Configure 802.1X authentication:** On Switch A, enable 802.1X authentication and set the user ports to controlled ports (Gi 0/3 and Gi 0/4). After configuration, the user will need to pass Dot1x authentication before accessing the Internet.
- **Configure the assignment of privilege-based IP address:** On Switch A, enable DHCP Option dot1x and configure IP authorization mode as DHCP server mode. After configuration, the DHCP server can allocate IP addresses according to user's privilege.

Note

1. Relevant configurations of 802.1X are detailed in the *802.1X Configuration*.

Note

2. The implementation of this example also needs the configuration of SAM Server and the DHCP server. For relevant details, see the relevant documents.

Configuration Steps

Configuring Switch A

Configure the address of the user gateway and the address of server interface.

! Configure the VLANs corresponding to Gi 0/3 and Gi 0/4 and configure the SVI corresponding to each VLAN.

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)#interface gigabitEthernet 0/3
Ruijie(config-if-GigabitEthernet 0/3)#switchport access vlan 10
Ruijie(config-if-GigabitEthernet 0/3)#exit
Ruijie(config)#interface gigabitEthernet 0/4
Ruijie(config-if-GigabitEthernet 0/4)#switchport access vlan 20
Ruijie(config-if-GigabitEthernet 0/4)#exit
Ruijie(config)#interface vlan 10
Ruijie(config-if-VLAN 10)#ip address 192.168.10.1 255.255.255.0
Ruijie(config-if-VLAN 10)#exit
Ruijie(config)#interface vlan 20
Ruijie(config-if-VLAN 20)#ip address 192.168.20.1 255.255.255.0
Ruijie(config-if-VLAN 20)#exit
```

! Configure the interface address of the DHCP server and SAM Server.

```
Ruijie(config)#interface gigabitEthernet 0/1
Ruijie(config-if-GigabitEthernet 0/1)#no switchport
Ruijie(config-if-GigabitEthernet 0/1)#ip address 10.1.1.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 0/1)#exit
Ruijie(config)#interface gigabitEthernet 0/2
Ruijie(config-if-GigabitEthernet 0/2)#no switchport
Ruijie(config-if-GigabitEthernet 0/2)#ip address 10.1.2.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 0/2)#exit
```

Configure relevant features of DHCP Relay.

! Configure the address of DHCP server as 10.1.1.2/24 and enable DHCP service.

```
Ruijie(config)#ip helper-address 10.1.1.2
Ruijie(config)#service dhcp
```

! Enable DHCP Option dot1x.

```
Ruijie(config)#ip dhcp relay information option dot1x
```

Configure 802.1X relevant features.

! Enable AAA and configure the address of Radius Server as 10.1.2.2/24; configure Radius Key as "ruijie".

```
Ruijie(config)#aaa new-model
Ruijie(config)#radius-server host 10.1.2.2
Ruijie(config)#radius-server key ruijie
```

! Create Dot1x authentication method list named "d1x" and configure Dot1x to apply such authentication method list.

```
Ruijie(config)#aaa authentication dot1x d1x group radius
Ruijie(config)#dot1x authentication d1x
```

! Configure ports Gi 0/3 and Gi 0/4 as controlled ports.

```
Ruijie(config)#interface range gigabitEthernet 0/3-4
Ruijie(config-if-range)#dot1x port-control auto
Ruijie(config-if-range)#exit
```

! Configure IP authorization mode as DHCP server mode.

```
Ruijie(config)#aaa authorization ip-auth-mode dhcp-server
```

Verification

Verify configurations of devices.

! Configurations of Switch A

```
Ruijie#show running-config
!
aaa new-model
!
aaa authorization ip-auth-mode dhcp-server
aaa authentication dot1x d1x group radius
```

```
!
vlan 10
!
vlan 20
!
service dhcp
ip helper-address 10.1.1.2
!
ip dhcp relay information option dot1x
!
radius-server host 10.1.2.2
radius-server key ruijie
!
dot1x authentication d1x
interface GigabitEthernet 0/1
 no switchport
 no ip proxy-arp
 ip address 10.1.1.1 255.255.255.0
!
interface GigabitEthernet 0/2
 no switchport
 no ip proxy-arp
 ip address 10.1.2.1 255.255.255.0
!
interface GigabitEthernet 0/3
 switchport access vlan 10
 dot1x port-control auto
!
interface GigabitEthernet 0/4
 switchport access vlan 20
 dot1x port-control auto
!
interface VLAN 10
 no ip proxy-arp
 ip address 192.168.10.1 255.255.255.0
!
interface VLAN 20
 no ip proxy-arp
 ip address 192.168.20.1 255.255.255.0
!
```

Configuring NTP

Understanding NTP

Network Time Protocol (NTP) is designed for time synchronization on network devices. With its clock source or the server. Moreover, NTP can provide time correction (the time difference is less than one millisecond on the LAN and dozens of milliseconds on the WAN, compared with the standard time) and prevent attacks using encryption and confirmation.

To provide accurate Coordinated Universal Time (UTC), NTP needs an accurate clock source from the atom clock, observatory, satellite or Internet.

To prevent the time server from malicious attacks, NTP uses an authentication mechanism is used to check whether the time synchronization request really comes from the declared server, and check the return path, thus providing the protection of anti-interference.

Ruijie switches support the NTP client and server. That is, the switch can synchronize the time from the time server and work as the time server (only in unicast server mode) to synchronize the time of other switches.

Configuring NTP

This chapter describes how to configure the NTP client and server.

Configuring the Global NTP Authentication Mechanism

Ruijie NTP client supports encrypted communication with the NTP server using key encryption.

Configure the encrypted communication between the NTP client and the NTP server as follows: Step 1, Authenticate the NTP client and configure the key globally; Step 2, Configure the trusted key for the NTP server. To initiate the encrypted communication with the NTP server, you need to set the authentication key for the NTP server in addition to performing Step 1.

By default, the NTP client does not use the global security authentication mechanism and the communication will not be encrypted. To enable encrypted communication, you need to enable the global security authentication, configure other global keys and set an encryption key for the server.

Use the following commands in global configuration mode to configure the global security authentication mechanism.

Command	Function
ntp authenticate	Configures the global NTP security authentication mechanism.
no ntp authenticate	Disables the global NTP security authentication mechanism.

To verify the packet, use the trusted key specified by the **ntp authentication-key** or **ntp trusted-key** command.

Configuring the Global NTP Authentication Key

The next step to configure the global security authentication for the NTP is to set the global authentication key.

Each key is identified by a globally unique key-id. You can use the **ntp trusted-key** command to set the key corresponding to the key-id as a global trusted key.

Use the following commands in global configuration mode to specify a global authentication key.

Command	Function
ntp authentication-key *key-id* **md5** *key-string* [*enc-type*]	Specifies a global authentication key. *key-id*: sets the parameter in the range 1 to 4294967295. *key-string*: sets the parameter to any values. *enc-type*: sets the parameter to **0** or **7**.
no ntp authentication-key *key-id* **md5** *key-string* [*enc-type*]	Removes a global authentication key.

The global authentication key takes effect after being configured as a global trusted key.

Caution Ruijie's current NTP version supports up to 1024 authentication keys, but only one key can be set for each server for encrypted communication.

Configuring the Global NTP Trusted key ID

The last step is to set a global authentication key as a global trusted key. Only by this trusted key you can send encrypted data and check the validity of the packet.

Use the following commands to specify a global trusted key in global configuration mode.

Command	Function
ntp trusted-key *key-id*	Specifies a global trusted key ID.
no ntp trusted-key *key-id*	Removes a global trusted key ID.

The three steps are the basis of implementing the security authentication mechanism. To initiate encrypted communication between the NTP client and the NTP server, a trusted key must be set for the corresponding server.

Caution When a global authentication key is removed, its trusted information is also removed.

Configuring the NTP Server

No NTP server is configured by default. Ruijie’s client can simultaneously interact with up to 20 NTP servers, and one authentication key can be set for each server to initiate encrypted communication with the NTP server after relevant settings of global authentication and keys are completed.

NTP version 3 is used in communication with the NTP server by default. NTP version 3 enables you to specify the source interface for sending the NTP packet and configure that the NTP packet from the relevant server can only be received on the sending interface at the same time.

Use the following commands to configure the NTP server in global configuration mode.

Command	Function
ntp server ip-addr [**version** *version*] [**source** *if-name number*] [**key** *keyid*] [**prefer**]	Configures the NTP server. version (NTP version number): sets the parameter in the range 1 to 3 if-name (interface type): sets the parameter to **Aggregateport, Dialer GigabitEthernet**, **Loopback, Multilink, Null, Tunnel, Virtual-ppp, Virtual-template,** or **VLAN.** keyid: sets the parameter in the range 1 to 4294967295.
no ntp server *ip-addr*	Removes the NTP server.

The NTP client can initiate the encrypted communication with the NTP server only when the global security authentication and key setting mechanisms are completed and the trusted key for communicating with the server is set. To this end, the NTP server should have the same trusted key.

Disabling the Function of Receiving the NTP Packet on the Interface

Use this command to disable the function of receiving the NTP packet on the interface for time synchronization, which is available to the NTP client by default.

Caution

This command takes effect only for the interface whose IP address can be configured to receive and send packets.

Use the following commands to disable the interface to receive the NTP packet in interface configuration mode.

Command	Function
interface interface-type number	Enters interface configuration mode.
ntp disable	Disables the function of receiving NTP packets on the interface.

To enable the function of receiving NTP packets on the interface, use the **no ntp disable** command in interface configuration mode.

Enabling or Disabling NTP

Use the **no ntp** command to disable the NTP synchronization service, stop the time synchronization, and clear relevant information of NTP configuration.

The NTP function is disabled by default, but may be enabled as long as the NTP server is configured.

Use the following commands to disable or enable the NTP in global configuration mode:

Command	Function
ntp authenticate or ntp server *ip-addr* [version *version*] [source *if-name number*] [*key keyid*] [*prefer*]	Enables NTP.
no ntp	Disables NTP.

Configuring the NTP Real-time Synchronization

To improve accuracy, eight consecutive packets are synchronized for the first synchronization between the client and the server. Follow-up NTP synchronization occurs automatically every one minute. To manually implement real-time synchronization during the auto-synchronization interval, you can use this command.

Use the following commands to configure the NTP real-time synchronization in global configuration mode.

Command	Function
ntp synchronization	Enables the NTP real-time synchronization.
no ntp synchronization	Disables the NTP real-time synchronization.

The synchronization is set to be implemented every 30 minutes on Ruijie's client system. New servers will trigger the real-time synchronization, which is also be implemented when this command is used during the synchronization interval. However, the command is invalid during the auto-synchronization.

The command of disabling the real-time synchronization and the one that disables NTP can be used to end the time synchronization (during the synchronization) or disable the synchronization function (between the synchronization interval). The difference lies in that the NTP-disabling command disables the NTP synchronization as well as clears related NTP configuration.

Note

The NTP real-time synchronization is supported only by some products. The **ntp synchronize** command cannot be executed on products that do not support this function.

Configuring the NTP Update-Calendar

Use this command to enable the NTP client to update the calendar using the clock time synchronized from an external clock source.

Use the following commands to configure the NTP update-calendar in global configuration mode.

Command	Function
ntp update-calendar	Configures the update-calendar.
no ntp update-calendar	Disables the update-calendar.

The update-calendar is not configured by default. After configuration, the NTP client updates the calendar when the time synchronization of external clock source is successful. It is recommended to enable this function for keeping the accurate calendar.

Setting the NTP Master

Use this command to set the local clock as the NTP master (the reference source of the local clock is reliable), providing the synchronized time for other devices.

Generally, the local system synchronizes the time from the external clock source directly or indirectly. However, if the time synchronization of the local system fails for the network connection trouble, use the command to set the reliable reference source of the local time, providing the synchronized time for other devices.

Once set, the system time can not be synchronized to the clock source with higher stratum.

Note

NTP uses stratum to describe the hops between the device and the authorization clock. A time server with 1 stratum shall have a directly connected atomic clock or radio wave clock. A time server with the 2 stratums obtains time from the stratum 1 server and a time server with 3 stratums obtains time from the stratum 2 server and so on. Therefore, the clock source with the lowest stratum value is more precise than others.

Use the following commands to configure the NTP master in global configuration mode:

Command	Function
ntp master [*stratum*]	Sets the local time as the NTP master and specifies the corresponding stratum. The time stratum is in the range 1 to 15. The default value is 8.
no ntp master	Cancels the NTP master setting.

The following example shows how to set the reliable reference source of the local time and set the time stratum to 12:

```
Ruijie(config)# ntp master 12
```

Caution

Be careful when using this command. Using this command to set the local time as the master (in particular, specify a lower stratum value), is likely to cover the effective clock source. If multiple devices in the same network use this command, time synchronization instability may occur due to time difference between the devices.

Caution

In addition, before using this command, if the system has never been synchronized with an external clock source, it is necessary to manually calibrate the system clock to prevent too much bias. (For more information, see the section about system time configuration in the *Basic Host management Configuration Guide*.)

Configuring the Access Control Privilege of NTP Service

The NTP service access control function provides a minimal security measure (a more secure way is to use the NTP authentication mechanism). By default, no NTP access control rules are configured in the system.

Use the following commands to set the NTP services access control privilege in global configuration mode.

Command	Function
ntp access-group { **peer** | **serve** | **serve-only** | **query-only** } *access-list-number* | *access-list-name*	Sets the access control privilege of the local service.
no ntp access-group { **peer** | **serve** | **serve-only** | **query-only** } *access-list-number* | *access-list-name*	Cancels the settings of access control privilege of the local service.

peer: allows the time requests and control queries for the local NTP service as well as the time synchronization between the local device and the remote system (full access privilege).

serve: allows the time requests and control queries for the local NTP service, not the time synchronization between the local device and the remote system.

serve-only: allows the time requests for the local NTP service.

query-only: allows the control queries for the local NTP service.

access-list-number: indicates the IP access control list label in the range of 1 to 99 and 1300 to 1999. For how to create IP access control list, see the *Access Control List Configuration Guide*.

access-list-name: indicates the IP access control list name. For how to create IP access control list , see the *Access Control List Configuration Guide*.

When an access request arrives, the NTP service matches the rules from the smallest to the largest access restriction, and the first matched rule shall prevail. The matching order is *peer*, *serve*, *serve-only*, and *query-only*.

⚠ Caution

The control query function (used by the network management device to control the NTP server, such as setting the leap second mark or monitoring the working state) is not supported in the current system. Although it matches with the order in accordance with the preceding rules, requests related to the control query function are not supported.

If you do not configure any access control rules, all accesses are allowed. Once the access control rules are configured, only the rule that allows access can be carried out.

The following example shows how to allow the peer device in acl1 to control the query, request for and synchronize the time with the local device; and limit the peer device in acl2 to request the time for the local device.

```
Ruijie(config)# ntp access-group peer 1
Ruijie(config)# ntp access-group serve-only 2
```

Showing NTP Information

Debugging NTP

Use this command to debug NTP for diagnosis and troubleshooting,.

Use the following commands to enable or disable the function of debugging NTP in privilege mode.

Command	Function
debug ntp	Enables the debugging function.
no debug ntp	Disables the debugging function.

Showing NTP Information

Use the **show ntp status** command in privilege mode to show the current NTP information.

Use the following command to show the NTP status information in privilege mode.

Command	Function
show ntp status	Shows the current NTP information.

This command be used to print the shown information only when the relevant communication server is configured.

```
Ruijie# show ntp status
Clock is synchronized, stratum 9, reference is 192.168.217.100
nominal freq is 250.0000 Hz, actual freq is 250.0000 Hz, precision is 2**18
reference time is AF3CF6AE.3BF8CB56 (20:55:10.000 UTC Mon Mar 1 1993)
clock offset is 32.97540 sec, root delay is 0.00000 sec
root dispersion is 0.00003 msec, peer dispersion is 0.00003 msec
```

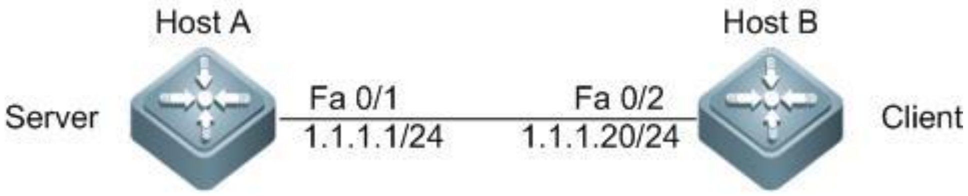
Note

stratum indicates the level of current clock;
reference indicates the address of the server used for synchronization;
freq indicates the clock frequency of current system;
precision indicates the precision of current system clock;
reference time indicates the UTC time of reference clock on the synchronization server;
clock offset indicates the offset of current clock;
root delay indicates the delay of current clock,
root dispersion indicates the precision of top server;
peer dispersion indicates the precision of synchronization server.

Typical NTP Configuration Examples

Configuring NTP Client or Server Mode

Topological Diagram



NTP client or server model

Application Requirements

On Host A, the local clock is configured as the NTP master clock, with the clock stratum being 12;
Host B is configured as the NTP client and Host A is specified as the NTP server;
The hardware clock of Host B shall be synchronized as well.

Configuration Tips

NTP server

Generally, the local system will directly or indirectly synchronize with the external clock sources. However, the local system may not be able to synchronize with the external clock sources due to the network connection failure. In this case, you can use the **ntp master** command to configure the local clock as NPT master to synchronize time to other devices.

NTP client

Configure the NTP server

By configuring the NTP update-calendar, the NTP client can use the clock value synchronized from external clock sources to update its calendar for accuracy.

Configuration Steps

Configuration of the NTP server

! Configure the NTP master. Configure local clock as the trusted reference clock source, with the clock stratum being 12.

```
HostA(config)# ntp master 12
```

Configuration of the NTP client

! Configure Host A as the NTP server.

```
HostB(config)#ntp server 1.1.1.1
```

! Configure NTP hardware clock update

```
HostB(config)# ntp update-calendar
```

Verification

Verify the time before configuring NTP synchronization.

! Verify the time of reference clock source.

```
HostA#show clock
17:12:48 UTC Tue, Sep 8, 2009
```

! Verify the time of client before synchronization.

```
HostB#show clock
12:01:10 UTC Sat, Jan 1, 2000
```

! Verify the NTP status of client before synchronization.

```
HostB(config)#show ntp status
Clock is unsynchronized, stratum 16, no reference clock
nominal freq is 250.0000 Hz, actual freq is 250.0000 Hz, precision is 2**0
reference time is 0.0 (00:00:00.000 UTC Thu, Jan 1, 1970)
clock offset is 0.00000 sec, root delay is 0.00000 sec
root dispersion is 0.00000 msec, peer dispersion is 0.00000 msec
```

The output shows that the time hasn't been synchronized.

After configuring NTP synchronization, show NTP configurations. Key points: the NTP server address and stratum.

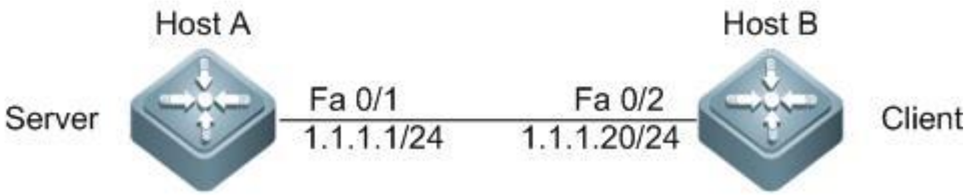
The following log information will be displayed on CLI interface:

```
*Sep  8 18:10:37: %SYS-6-CLOCKUPDATE: System clock has been updated to 18:10:37 UTC Tue Sep
8 2009.
HostB#show ntp status
Clock is synchronized, stratum 13, reference is 1.1.1.1
nominal freq is 250.0000 Hz, actual freq is 250.0000 Hz, precision is 2**24
reference time is CE511CC9.37EB5B2D (18:11:21.000 UTC Tue, Sep 8, 2009)
clock offset is -0.00107 sec, root delay is 0.00000 sec
root dispersion is 0.00002 msec, peer dispersion is 0.00002 msec
```

The output shows that the NTP client has connected to the server and the time of Host B has been synchronized with the time of Host A, with the stratum level being higher than that of Host A by 1 (i.e., 13).

Configuring NTP Client or Server Mode with Authentication

Topological Diagram



NTP client/server model

Application Requirements

On Host A, the local clock is confiugred as the NTP master clock, with the clock stratum being 12;
Host B is configured as the NTP client and Host A is specified as the NTP server;
The authentication mechanism is enabled to prevent illegal users from maliciously attacking the clock server.

Configuration Tips

Configuring NTP server/client authentication will involve the following steps:

Enable NTP global authentication.
Configure the key for NTP global authentication and the corresponding key ID.
Specify NTP global trusted key ID.

The authentication key used by NTP client to communicate with NTP server must be consistent with the corresponding Key ID.

Configuration Steps

Configuration of the NTP server

Step 1: Configure the NTP master. Configure the local clock as the trusted reference clock source, with the clock stratum being 12;

```
HostA(config)#ntp master 12
```

Step 2: Configure NTP authentication;

! Enable NTP global authentication.

```
HostA(config)# ntp authenticate
```

! Configure the NTP global authentication key as **helloworld** and the corresponding key ID as **6**.

```
HostA(config)# ntp authentication-key 6 md5 helloworld
```

! Specify **6** as the NTP global trusted key ID

```
HostA(config)# ntp trusted-key 6
```

Configuration of the NTP client

Step 1: Configure NTP authentication;

! Enable NTP global authentication.

```
HostB(config)# ntp authenticate
```

! Configure NTP global authentication key as **helloworld** and the corresponding key ID as **6**.

```
HostB(config)# ntp authentication-key 6 md5 helloworld
```

! Specify **6** as the NTP global trusted key ID.

```
HostB(config)# ntp trusted-key 6
```

! Configure Host A as the NTP server and set the key ID for communicating with this server as **6**.

```
HostB(config)# ntp server 1.1.1.1 key 6
```

Verification

Verify the configurations of NTP server. Key points: the NTP master clock configuration, NTP server's IP address, and authentication related configurations.

```
HostA#show run
!
interface fastEthernet 0/1
ip address 1.1.1.1 255.255.255.0
!
ntp authentication-key 6 md5 07360623191d300a004609 7
ntp authenticate
ntp trusted-key 6
ntp master 12
!
```

Verify the configurations of NTP client. Key points: the IP address and key ID of NTP server, and authentication related configurations.

```
HostB #show run
!
interface fastEthernet 0/2
 ip address 1.1.1.20 255.255.255.0
!
ntp authentication-key 6 md5 141a4f012d1d3c23174905 7
ntp authenticate
ntp trusted-key 6
ntp server 1.1.1.1 key 6
!
```

After proper configurations, the following log information will be displayed on the CLI:

```
*Sep  9 11:31:29: %SYS-6-CLOCKUPDATE: System clock has been updated to 11:31:29 UTC Wed Sep 9 2009.
```

The log information indicates that the clock of HostB (NTP client) has been updated.

Verify NTP server status.

```
HostA #show ntp status
Clock is synchronized, stratum 12, reference is 127.127.1.1
nominal freq is 250.0000 Hz, actual freq is 250.0000 Hz, precision is 2**24
reference time is CE521261.E52DECA2 (11:39:13.000 UTC Wed, Sep 9, 2009)
clock offset is 0.00000 sec, root delay is 0.00000 sec
root dispersion is 0.00002 msec, peer dispersion is 0.00002 msec
```

Verify NTP client status. Key points: the NTP server address and stratum.

```
HostB# show ntp status
Clock is synchronized, stratum 13, reference is 1.1.1.1
nominal freq is 250.0000 Hz, actual freq is 250.0000 Hz, precision is 2**24
reference time is CE5212A1.E5D712A0 (11:40:17.000 UTC Wed, Sep 9, 2009)
clock offset is -0.00005 sec, root delay is 0.00000 sec
root dispersion is 0.00002 msec, peer dispersion is 0.00002 msec
```

The output shows that the NTP client has successfully connected to the server and the time of Host B has been synchronized with that of Host A, with the stratum level being higher than that of Host A by 1 level (i.e., 13).

Configuring SNTP

Overview

Network Time Protocol (NTP) is designed for time synchronization on network devices. Another protocol, the Simple Network Time Protocol (SNTP) can also be used to synchronize the network time.

NTP can be used across various platforms and operating systems, provide precise time calculation (1-50 ms precision), and prevent from latency and jitter in the network. NTP also provides the authentication mechanism with a high security level. However, the NTP algorithm is complicated and demands delicate systems.

As a simplified version of NTP, SNTP simplifies the algorithm of time calculation while maintains great performance, with the precision about 1s.

The SNTP client is totally compatible with the NTP server due to the consistency between the SNTP and NTP packets.

Understanding SNTP

SNTP works in client/server mode. The standard server system time is set by receiving the GPS signal or the atomic clock. The client obtains its accurate time from the service time accessing the server regularly, and adjusts its system clock to synchronize the time.

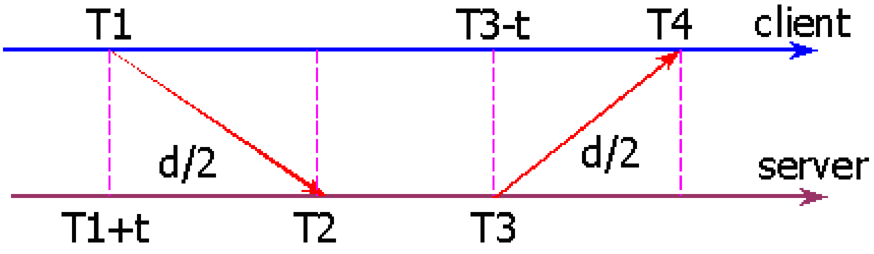


Figure-1

Originate Timestamp	T1	Time request sent by client
Receive Timestamp	T2	Time request received at server
Transmit Timestamp	T3	Time reply sent by server
Destination Timestamp	T4	Time reply received at client

T1: Time request sent by client (refer to the client time) with the mark “Originate Timestamp”;

T2: Time request received at server (refer to the server time) with the mark “Receive Timestamp”;

T3: Time reply by server (refer to the server time) with the mark “Transmit Timestamp”;

T4: Time reply received at client (refer to the client time) with the mark “Destination Timestamp”.

T: Time offset between the server and the client d: Round-tour time between the server and the client

The following formula calculates the time:

```
∵ T2 = T1 + t + d / 2;

∴ T2 - T1 = t + d / 2;

∵ T4 = T3 - t + d / 2;

∴ T3 - T4 = t - d / 2;


∴ d = (T4 - T1) - (T3 - T2);

t = ((T2 - T1) + (T3 - T4)) / 2;
```

Then, according to the value of *t* and *d*, the SNTP client gets the current time: T4 + t.

Configuring SNTP

This chapter describes how to configure SNTP.

Default Configuration

The following table describes the default SNTP configurations.

Function	Default
SNTP state	Disabled
IP address for the NTP server	0
SNTP Sync Interval	1800 seconds
Local Time-zone	GMT + 8

Enabling SNTP

Enter privileged mode and perform the following steps to enable the SNTP:

Enter global configuration mode:

```
Ruijie# config
```

Enable the SNTP and synchronize the time once immediately. The time will be immediately synchronized if this command is entered and regular synchronization is unnecessary. (in order to prevent frequent time synchronization, the sync-interval must not be less than 5 seconds)

```
Ruijie(config)# sntp enable
Return to privileged mode:
Ruijie(config)# End
```

Show the current configuration:

```
Ruijie# show running-config
```

Save the configuration:

```
Ruijie# copy running-config startup-config
```

To disable the SNTP, use the **no sntp enable** command.

Configuring the IP address of the SNTP Server

The SNTPclient is totally compatible with the NTP server due to the consistency between SNTP and NTP packets. There are many NTP servers in the network, and you can choose one with less latency.

For the detailed NTP server ip addresses, please log on to http://www.time.edu.cn/ or http://www.ntp.org/. For example, 192.43.244.18 (time.nist.gov).

Enter privileged mode and perform the following steps to specify an IP address for the SNTP server:

Enter global configuration mode:

```
Ruijie# config
```

Specify the IP address for the SNTP server.

```
Ruijie(config)# sntp server <ip-addr>
```

Return to privileged mode:

```
Ruijie(config)# End
```

Show the current configuration:

```
Ruijie# show running-config
```

Save the configuration:

```
Ruijie# copy running-config startup-config
```

Configuring the SNTP Synchronization Interval

To adjust the time regularly, you need to set the synchronization interval for SNTP client to access the NTP server SNTP client regularly. Perform the following steps to set the sync interval for the device and the NTP server:

Enter global configuration mode:

```
Ruijie# config
```

Configure the SNTP sync interval, in second.

Interval range: 60-65535s; Default value: 1800s.

```
Ruijie(config)# sntp interval <seconds>
```

Return to privileged mode:

```
Ruijie(config)# End
```

Show the current configuration:

```
Ruijie# show running-config
```

Save the configuration:

```
Ruijie# copy running-config startup-config
```

Caution

The synchronization interval configuration cannot take effect immediately unless you execute the **sntp enable** command immediately after configuring the synchronization interval.

Configuring the Local Time Zone

Greenwich Mean Time (GMT) is obtained through the SNTP communication. To obtain the accurate local time, you need to set the local time to adjust the mean time.

Enter global configuration mode:

```
Ruijie# config
```

Configure the time-zone, ranging from GMT-23 to GMT+23, wherein “-” indicates western area, “+” indicates eastern area. For example "8" indicates the 8th eastern time zone, "-8" indicates the 8th western time zone and “0” indicates Greenwich mean time. Universal Time Coordinated (UTC) is the default time zone name and the default value is **0**.

```
Ruijie(config)# clock timezone <time-zone>
```

Return to privileged mode:

```
Ruijie(config)# end
```

Show the current configuration:

```
Ruijie# show running-config
```

Save the configuration:

```
Ruijie# copy running-config startup-config
```

To restore the local time-zone to the default, use the command **no clock time-zone**.

Showing SNTP Information

The procedure is as follows:

Show related SNTP parameters:

```
Ruijie# show sntp
```

Use the **show sntp** command to show configured SNTP parameters:

```
Ruijie# show sntp
SNTP state              : ENABLE          //to view whether SNTP is enabled or not
SNTP server             : 192.168.4.12   //NTP Server
SNTP sync interval       : 60           //SNTP sync interval
Time zone             : +8               //Local Time-zone
```

Configuring UDP-Helper

Understanding UDP-Helper

Overview

UDP-Helper relays and forwards User Datagram Protocol (UDP) broadcast packets. As a relay, UDP-Helper can convert the UDP broadcast packets into the unicast packets and then forward them to the specified destination server by configuring the destination server for the UDP broadcast packets to be forwarded.

Once enabled, UDP-Helper checks whether the destination UDP port number of the received packet matches the port number to be forwarded to. If yes, it modifies the destination IP address of packets as the IP address of the specified destination server, and sends the packet to the destination server in unicast form.

When UDP-Helper is enabled, the broadcast packets from Ports 69, 53, 37, 137, 138 and 49 are relayed and forwarded by default.

Note

The BOOTP/DHCP broadcast packet is relayed through the UDP Port 67 and 68 by the DHCP Relay module; therefore, the two ports cannot be configured as the relay port of UDP-Helper.

Configuring UDP-Helper

Default Configuration

The following table describes the default configuration.

Attribute	Default value
Relay and forwarding	Disabled
UDP port for relay and forwarding	Indicates that the UDP broadcast packets from Ports 69, 53, 37, 137, 138 and 49 are relayed and forwarded by default when UDP-Helper is enabled.
Destination server for delay and forward	None

Enable the Relay and Forward Function of UDP-Helper

Command	Function
Ruijie(config)# **udp-helper** **Enable**	Enables the relay and forward function of UDP broadcast packets. This function is disabled by default.

To disable this function ,sue the **no udp-helper enable** command.

Note This function is disabled by default.

Note When UDP-Helper is enabled, the broadcast packets from UDP Ports 69, 53, 37, 137, 138 and 49 are relayed and forwarded by default.

Note When UDP-Helper is disabled, all of the configured UDP ports including the default ports are cancelled.

Configuring the Destination Server for Relay and Forwarding

Command	Function
Ruijie(config-if)# **ip helper-address** *IP-address*	Configures the destination server to which the UDP broadcast packets are relayed and forwarded. By default, it is not configured.

To remove the destination server for relay and forwarding, use the **no ip helper-address** command.

Note At most 20 destination servers can be configured for an interface.

Note If the destination server for relay and forwarding is configured on a specified interface and UDP-Helper is enabled, the broadcast packets of the specified UDP port received from this interface will be sent to the destination server configured for this interface in unicast form.

Configuring the UDP Port for Relay and Forwarding

Command	Function
Ruijie(config)# **ip forward-protocol udp** [*port*]	Configures the UDP port for relay and forwarding. If only the UDP parameter is specified, the default port will be used for relay and forwarding; otherwise, the port can be configured if necessary. When UDP-Helper is enabled, the broadcast packets from Ports 69, 53, 37, 137, 138 and 49 are relayed and forwarded by default.

To disable the UDP port for relay and forwarding, use the **no ip forward-protocol udp** [*port*] command.

Note

The UDP port can be configured for relay and forwarding. Otherwise, the error prompts will appear only when the function of delay and forwarding is enabled for UDP-Helper and the destination server is configured for the relay and forwarding.

Note

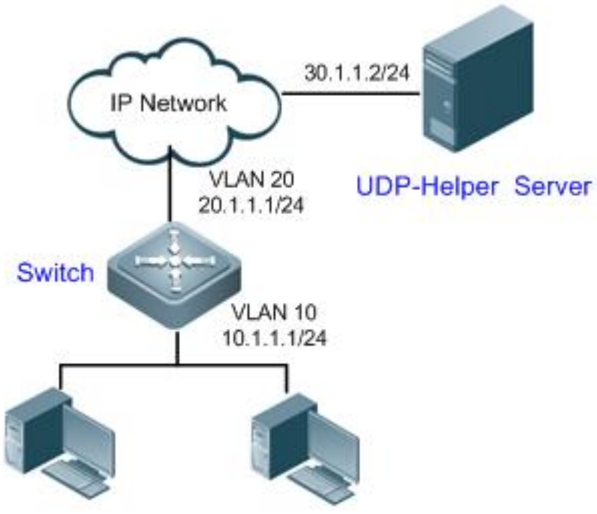
When the relay and forwarding function of UDP-Helper is enabled, the function of forwarding the broadcast UDP packets from the default ports 69, 53, 37, 137, 138 and 49 will be enabled immediately without any configuration.

At most 256 UDP ports are supported for relay and forwarding by the switch.

You can use both the **ip forward-protocol udp domain** and **ip forward-protocol udp** *53* commands to configure default ports.

Configuration Examples

Topology Diagram



Topology diagram for UDP-Helper configuration

Application Requirements

The network device can forward UDP broadcast packets with destination port being 1000 to the specified UDP-Helper server (with server IP being 30.1.1.2/24).

Configuration Tips

Configure UDP-Helper relay and forwarding as follows:

Enable UDP-Helper relay forwarding.

Configure the destination server of UDP-Helper relay forwarding.

Configure the destination port number of UDP broadcast packets for relay forwarding (in this example, UDP broadcast packets with destination port being 1000 are subject to relay forwarding; meanwhile, the device will by default forward UDP broadcast packets containing destination port numbers of 69, 53, 37, 137, 138 and 49).

Note

The UDP port for relay forwarding can only be configured after UDP-Helper relay forwarding is enabled and the destination server is configured; otherwise, error messages will be displayed.

After UDP relay forwarding is enabled, the device will immediately forward UDP broadcast packets containing the default port numbers of 69, 53, 37, 137, 138 and 49 without further configuration.

Configuration Steps

Before configuring relevant features of UDP-Helper, make sure that the route from the switch to the network segment of UDP-Helper server is reachable. The IP addresses configured on respective interfaces are shown in the topological diagram. Here we will introduce how to configure relevant features of UDP-Helper.

Step 1: Enable UDP-Helper relay forwarding on the network device

```
Ruijie(config)#udp-helper enable
```

Step 2: Configure the IP address for the destination server of UDP-Helper relay forwarding as 30.1.1.2 on fastEthernet 1/1.

```
Ruijie(config)# interface fastEthernet 1/1
Ruijie(config-if-VLAN 10)#ip address 10.1.1.1 255.255.255.0
Ruijie(config-if-VLAN 10)# ip helper-address 30.1.1.2
Ruijie(config-if-VLAN 10)#exit
```

Step 3: Configure the Switch to forward UDP broadcast packets carrying the destination port number 1000.

```
Ruijie(config)#ip forward-protocol udp 1000
```

Verification

Verify configurations of the switch. Key points: whether relay forwarding is enabled or not; IP address of relay server; destination port number carried in UPD broadcast packets requiring relay forwarding.

```
Ruijie#show run
!
udp-helper enable
!
ip forward-protocol udp 1000
!
interface fastEthernet 0/1
 ip address 10.1.1.1 255.255.255.0
!
interface fastEthernet 1/1
ip helper-address 30.1.1.2
```

```
 ip address 20.1.1.1 255.255.255.0
!
```

Verify whether relay forwarding has taken effect.

Step 1: Send UDP broadcast packets carrying the destination port number 999.

PC1 sends a UDP broadcast packet of the following format:

```
Src_mac:0000.0000.0001
Dst_mac:0xFFFFFFFFFFFF
Src_ip:1.0.0.3
Dst_ip:255.255.255.255
Dst_port:999
PC2 acts as UDP-Helper server. Such packet is not received on PC2.
Step 2: Send UDP broadcast packets carrying the destination port number of 1000.
PC1 sends a UDP broadcast packet of the following format:
Src_mac:0000.0000.0001
Dst_mac:0xFFFFFFFFFFFF
Src_ip:1.0.0.3
Dst_ip:255.255.255.255
Dst_port:1000
PC2 acts as UDP-Helper server. Such packet is received on PC2. The destination IP address of packet is 30.1.1.2, and the data contained are the same as the packets sent.
```

Step 3: Send UDP broadcast packets with destination port number 69, 53, 37, 137, 138 or 49.

PC1 sends a UDP broadcast packet of the following format (taking destination port number of 69 as the example):

```
Src_mac:0000.0000.0001
Dst_mac:0xFFFFFFFFFFFF
Src_ip:1.0.0.3
Dst_ip:255.255.255.255
Dst_port:69
```

PC2 acts as UDP-Helper. Such packet is received on PC2. The destination IP address of packet is 30.1.1.2, and the data contained are the same as the packets sent.

From the verification output, we can learn that the switch has successfully forwarded UDP broadcast packets with the user-defined destination port number (destination port number 1000 and default numbers of 69, 53, 37, 137, 138, and 49) to the specified UDP-Helper server.

Configuring URPF

Understanding URPF

Overview

Recently, frequent Denial of Service (DOS) attacks caused by forged source address are causing many troubles to Internet Service Providers (ISPs) and network maintenance.

Figure 1 shows a common scenario of using a forged source address to perform DOS attacks.

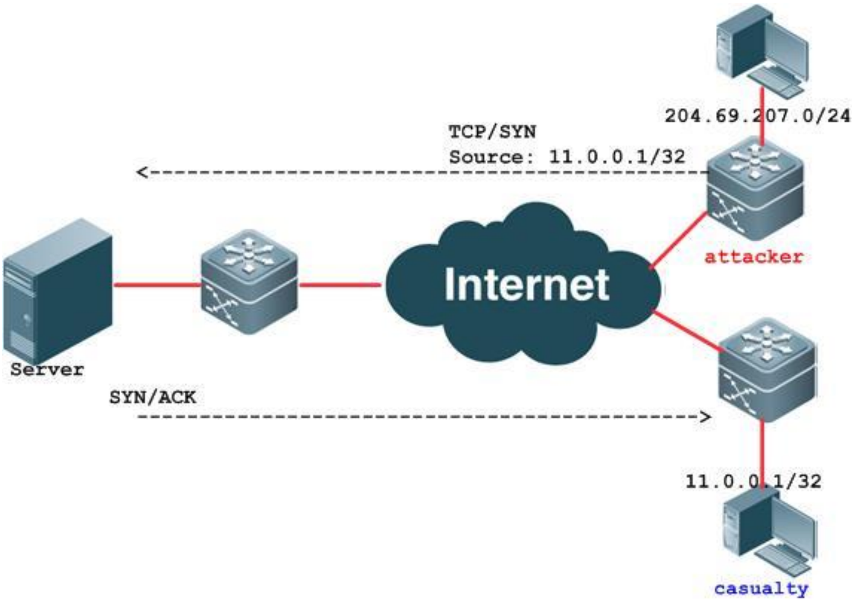


Figure 1 Scenario of source-address-based attacks

The attacker initiates attacks by sending packets with the forged source address 11.0.0.1, making the server send excessive SYN/ACK packets to the host unrelated to this attack, and the host with the real source address is also affected. What's worse, if the network administrator identifies that this address is related to the attack on the network and discards all data streams from this source address; therefore, the denial of service to the source address occurred.

Unicast Reverse Path Forwarding (URPF) well addresses the preceding problem.

It is known that during packet forwarding, the forwarding table is looked up according to the destination address contained in the IP packet received, and the packet is forwarded according to the entry found in the forwarding table. URPF will look up the forwarding table for the forwarding entry according to source address and ingress interface of the incoming packet. If the forwarding entry is not found in the forwarding table, the packet will be discarded; if the egress interface specified in the forwarding table doesn't match with the ingress interface of the packet, the packet will also be discarded. Otherwise, the packet will be forwarded.

URPF can protect the network by intercepting source address spoofing attacks.

Features of URPF

URPF strict mode

Conventional URPF technical requirements: URPF will look up the forwarding table for the forwarding entry according to the source address and ingress interface of the incoming packet. If the forwarding entry is not found in the forwarding table, the packet will be discarded; if the egress interface specified in the forwarding table doesn't match with the ingress interface of the packet, the packet will also be discarded. This requires that the "ingress interface of the packet received must be the egress interface of the route reaching this source address". We call such URPF check mode as URPF strict mode.

Note

URPF strict mode is generally deployed on the point-to-point interface, and the data streams from both directions need to pass this point-to-point interface.

URPF loose mode

URPF strict mode has its limitations, and is particularly not applicable to the asymmetrical routing environment and multi-homed network environment.

To implement the network flow control and routing policy, asymmetrical routing is a common seen network application. In asymmetrical routing, URPF strict mode will result in the loss of data streams. Figure 2 shows an example of asymmetrical routing. If G1/2 on R1 enables URPF strict mode and receives packets from the network segment of 192.168.20.0/24, URPF check will indicate the interface of G1/1 and the packet will not be able to pass URPF check.

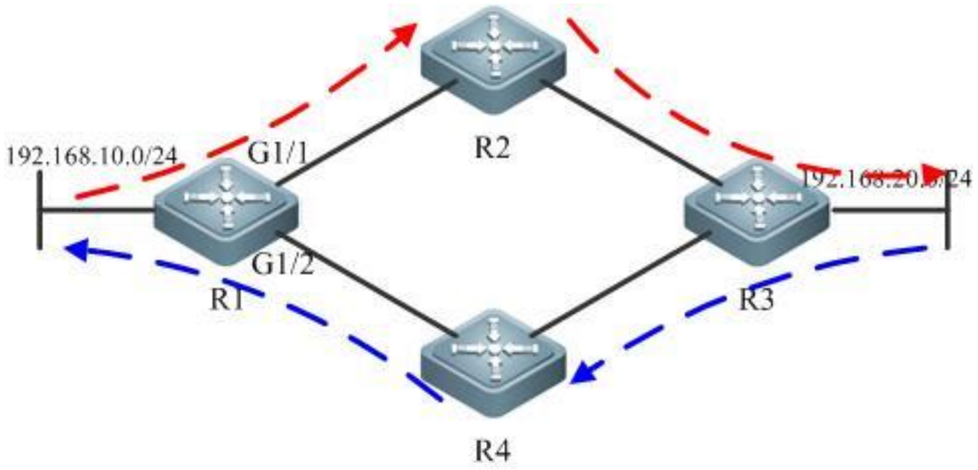


Figure 2 Asymmetrical routing

The multi-homed network is the common network application between a user and the ISP or between ISPs. As shown in Figure 3, user network A is simultaneously connected to multiple ISPs, and the incoming and outgoing streams are always asymmetrical. The figure shows that user network A is visiting user network B.

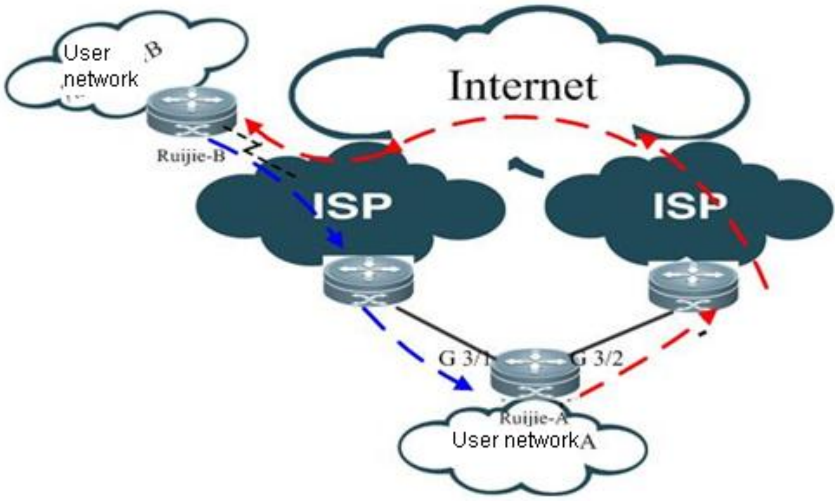


Figure 3 Multi-homed network

In the preceding two application scenarios, valid packets will be filtered if URPF strict mode is enabled. Therefore, URPF loose mode is introduced.

URPF loose mode: Reverse route check is conducted according to the source IP of incoming packet. As long as the route is found, the next-hop egress interface may unnecessarily be the interface that receives the packet. URPF loose mode well addresses the asymmetrical stream problem in the aforementioned asymmetrical routing application and multi-homed network.

URPF monitoring

To conveniently monitor the drop rate of packets after URPF is enabled, Ruijie devices initiatively inform the user of such drop rate via Syslog and Trap. The mechanism will be detailed in this section.

URPF monitoring has introduced the following concepts:

Drop rate: indicates the number of packets discarded due to URPF check within a specific period of time divided by this time. Unit: packets/second (pps).

Drop rate computation interval: indicates the time interval from the previous calculation to the recalculation of the drop rate.

Drop rate sampling interval: indicates the time interval for calculating the number of packets dropped. This value must not be smaller than the drop rate computation interval.

Drop rate notification threshold: indicates the maximum drop rate allowed. When the drop rate is higher than this threshold, the user will be notified via Syslog or Trap. The user may also adjust the drop rate notification threshold according to actual situations of the network.

Drop rate notify hold-down time: indicates the time interval between two successive notifications. The user may adjust the value according to actual situations of the network to avoid frequent Log printing or Trap sending.

The following figure shows two successive computations of the drop rate:

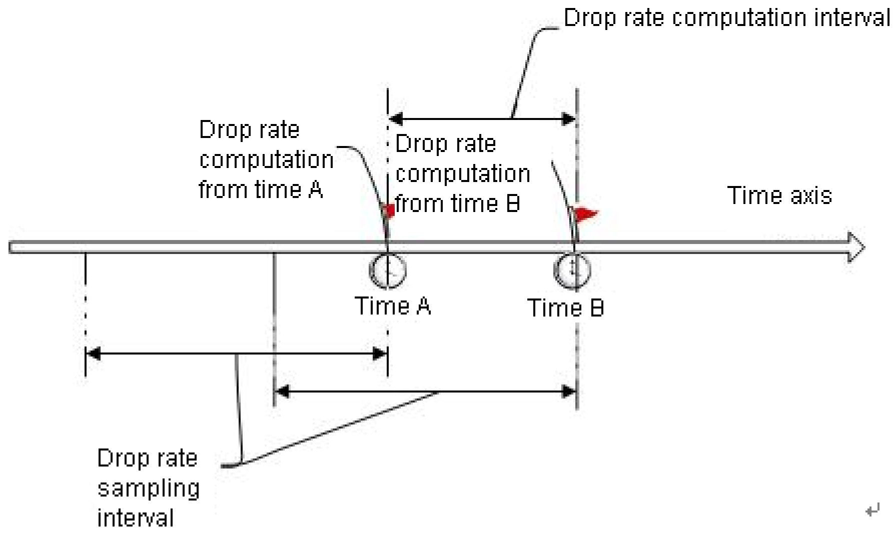

Drop rate computation interval
Drop rate computation from time A
Drop rate computation from time B
Time axis
Time A
Time B
Drop rate sampling interval

Figure 4 Two successive computations of drop rate

Main points of URPF monitoring:

Drop rate computation

After URPF function is enabled, within the time interval of "0-drop rate sampling interval (including the sampling time point reaching the drop rate)", the drop rate is computed by dividing the currently computed number of dropped packets by the time of URPF running. The subsequent method for drop rate computation: Computed the number of currently dropped packets at every drop rate computation interval, deduct the number of dropped packets before the drop rate sampling interval, and then divide the difference by drop rate sampling interval to obtain the current drop rate.

URPF allows interface-based and global-based drop rate computation and monitoring.

For configurations related to URPF monitoring, see the *Configuring URPF Drop Rate Notification*.

Working Principle

The working principle of URPF has been described in the section Features of URPF.

URPF functions can be applied to IPv4/IPv6 packets according to configurations. However, note that the following packets will not be subject to URPF check.

Destination addresses being multicast packets. URPF check is only applied to the source address with destination address being IPv4/IPv6 packet.
DHCP/BOOTP packets with source IP address being 0.0.0.0 and destination IP address being 255.255.255.255
IPv6 packets with source IP address being Link local addresses

Application Restrictions

Ruijie products that support URPF include:

1) Router products.
2) Switches that support URPF function:

S5750 V2.x product

Category B and C line cards and the M8600-MPLS line card of the S8600 series products

S12000EA line card

The URPF function supported by our router products has the following application restrictions:

Router products only support the URPF function during progress forwarding. If the fast forwarding is enabled on the interface (interface configuration mode: **ip ref**), the URPF function will be disabled.

Currently, the URPF function on routers cannot be applied to IPv6 packets.

The URPF function also has the following characteristics in switch and router products:

After the URPF function is enabled, a route whose source address is matched with a NULL interface will still be subject to URPF check.

After the URPF function is enabled, URPF enjoys precedence over the Access Control List (ACL, interface configuration mode: **ip access-group in**) during packet check.

If URPF strict mode is enabled, incoming packets with the source address being the address of the ingress interface will be discarded. If URPF loose mode is enabled, such packets will pass the interface.

Protocol Specification

Protocol specifications related to URPF include:

RFC 2827, Network Ingress Filtering: DDOS Attacks which employ IP Source Address Spoofing

RFC 3704, Ingress Filtering for Multi-homed Networks

Default Configurations

The following table describes the default configurations of URPF.

Function	Default setting
URPF global configuration mode	Disabled
URPF interface configuration mode	Disabled
URPF drop rate monitoring	Disabled
URPF drop rate computation interval	30 seconds
URPF drop rate sampling interval	URPF drop rate computation interval x 5
URPF drop rate notification threshold	1000 pps
URPF drop rate notify hold-down time	300 seconds
URPF Trap sent for drop rate notification	Disabled

Configuring URPF Functions

The following sections describe how to configure the basic functions of URPF:

Configuring URPF (Global Configuration Mode)

After the MPLS line card is inserted in the S8600 series switches, packets will be forwarded by the MPLS line card. In this case, you need to enable the URPF function according to the steps described in the following table.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# i**p verify unicast source reachable-via rx**	Enables the URPF function rx: uses strict mode to perform URPF check
Ruijie(config)# **end**	Exits global configuration mode
Ruijie# **show ip urpf**	Shows URPF configurations and statistics

To disable global URPF function, use the **no ip verify** command in global configuration mode.

⚠ Caution

The configuration of the URPF feature in global configuration mode takes effect on the S8600 series switches only after the MPLS line card is inserted. After the URPF function takes effect, it will enable URPF check on IPv4 packets.

⚠ Caution

The URPF function configured in global configuration mode supports only URPF strict mode. When used with equal-cost routing, it will switch to URPF loose mode.

⚠ Caution

URPF function configured in global configuration mode doesn't support URPF check with the default route.

⚠ Caution

The URPF function cannot be configured in global configuration mode and in interface configuration mode at the same time.

⚠ Caution

Note that it is not recommended to configure URPF globally if the S8600 series devices are directly connected to users' network segments. The URPF check fails and the packets are discarded if the S8600 series devices did not learn the Address Resolution Protocol (ARP) entry of a directly-connected user before packet forwarding.

Configuration example:

```
# Enable URPF in global configuration mode:
```

```
Ruijie# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# ip verify unicast source reachable-via rx
```

Configuring URPF (Interface Configuration Mode)

URPF is not enabled on the interface by default.

Use the following commands to enable the URPF function on the interface.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **interface** *interface-name*	Enters interface configuration mode.
Ruijie(config-if)# **ip verify unicast source reachable-via** { **any** | **rx** } [**allow-default** | *acl_name*]	Enables the URPF function. any: uses loose mode to perform URPF check. rx: uses strict mode to perform URPF check. allow-default: allows the use of default route to perform URPF check. *acl-name*: indicates the ACL number, supporting: 1 to 99 (IP standard access list) 100 to 199 (IP extended access list) 1300 to 1999 (IP standard access list, expanded range) 2000 to 2699 (IP extended access list, expanded range)
Ruijie(config-if)# **end**	Exits interface configuration mode and returns to privilege mode.
Ruijie# **show ip urpf interface** *interface-name*	Displays URPF configurations and statistics.

Note

By default, the default route is not used for URPF check; if required, the user can use the keyword *allow-default* to enable this function.

Note

By default, a packet failing in URPF check will be discarded. If the ACL (*acl-name*) is configured, such a packet will undergo ACL check after failing in URPF check. If ACL doesn't exist or the packet points to a deny ACE, such packet will be discarded. If the packet points to a permit ACE, the packet will be forwarded.

Caution

After this command is enabled, the S5700 V2.x switch and the S8600 and S12000 series switches will enable the URPF check on IPv4 and IPv6 packets at the same time, and the routers will enable the URPF check on IPv4 packets.

Caution

For switches, the URPF feature is supported only on the S5700 V2.x switch and the routed port and Layer 3 AP associated with category B line cards of S8600 series. The restrictions are as follows:

- The URPF function doesn't support the function of associating ACL options.
- The URPF function doesn't support the use of IPv6 routes with 65-to-127 bit prefix to perform URPF check.
- After URPF function is enabled, all packets received by the physical port of these interfaces will be subject to URPF check, thus expanding the range of URPF check. A typical application scenario is: If a packet received by Tunnel interface is received from the aforementioned physical port, this packet will also be subject to URPF check. If such an application scenario exists, be cautious when enabling URPF check.
- After URPF function is enabled, the route forwarding capacity of the device will be reduced by 50%.
- After URPF strict mode is enabled, if the packets received by the interface match with equal-cost routing during URPF check, it will switch into loose mode.
- The URPF function cannot take effect on interfaces of the S8600 series switches after the MPLS line card is inserted.

Caution

The URPF function cannot be configured in global configuration mode and in interface configuration mode at the same time.

Configuration example:

```
# Perform strict URPF check for packets received by interface
```

GigabitEthernet 0/21, with no need to use default route for URPF check.

```
Ruijie# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# interface gigabitEthernet0/21
Ruijie(config-if)# ip verify unicast source reachable-via rx
```

Configuring URPF Drop Rate Notification

To configure URPF drop rate notification, you must enable the URPF function first.

Use the following commands to configure URPF drop rate notification.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **ip verify urpf drop-rate compute interval** *seconds*	Configures URPF drop rate computation interval in seconds. The value range is 30 to 300. The default value is 30 seconds.

Ruijie(config)# ip verify urpf drop-rate notify **hold-down** *seconds*	Configures URPF drop rate notify hold-down time in seconds. The range is 30 to 300. The default value is 300 seconds.
Ruijie(config)# **interface** *interface-name*	Enters interface configuration mode.
Ruijie(config-if)# **ip verify urpf drop-rate notify**	Enables URPF drop rate monitoring.
Ruijie(config-if)# ip verify urpf notification threshold *rate-value*	Configures URPF drop rate notification threshold.in Packets Per Second (pps). The range is 0 to 4294967295. The default value is 1000 pps.
Ruijie(config-if)# **exit**	Exits interface configuration mode and returns to global configuration mode.
Ruijie(config)# **snmp-server enable traps urpf**	Sends Trap packets after the URPF drop rate has exceeded the notification threshold.
Ruijie(config)# **snmp-server host** {*host-addr*| *ipv6 ipv6-addr* } **traps** *word urpf*	Configures the host to receive URPF Trap.
Ruijie(config)# **end**	Exits global configuration mode.
Ruijie# **show ip urpf**	Shows URPF configurations and statistics.

Note

By default, the drop rate notification threshold is 1000 pps. The user may adjust the drop rate notification threshold as required.

Caution

Drop rate monitoring is only effective in interface configuration mode, and is not supported in global configuration mode.

Caution

In interface configuration mode, the drop rate will be computed according to the packets dropped by the interface after URPF check is enabled.

Configuration example:

# Perform strict URPF check for packets received by interface GigabitEthernet 0/21 with no need to use default route for URPF check. Monitor the URPF drop rate via SNMP Trap. Configure the drop rate notification threshold to 500 pps. Configure the SNMP host 192.168.12.219 to receive Trap packets.

```
Ruijie# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# ip verify urpf notification threshold 500
Ruijie(config)# snmp-server enable traps urpf
Ruijie(config)# snmp-server host 192.168.12.219 public urpf
Ruijie(config)# interface gigabitEthernet0/21
Ruijie(config-if)# ip verify unicast source reachable-via rx
```

Viewing URPF Configurations

Use the following command provided by URPF to show various configurations and statistics.

Command	Function
show ip urpf [**interface** *interface-name*]	Shows URPF configurations and statistics.

Use the following command provided by URPF to clear URPF statistics.

Command	Function
clear ip urpf [**interface** *interface-name*]	Clears the statistics about packets dropped in URPF check.

Typical URPF Configuration Example

Example of Strict Mode Configuration

Networking Requirements

Figure 5 shows the typical hierarchical network architecture.

The packets with forged source addresses may be transmitted from user PCs to the core-layer network. To avoid this, the source address attack packets need to be isolated on the access layer or distribution layer to eliminate invalid data on the aggregation-layer and core-layer network.

The preceding requirement can be satisfied by enabling URPF strict mode on the interface linking aggregation-layer devices and access-layer devices.

Networking Topology

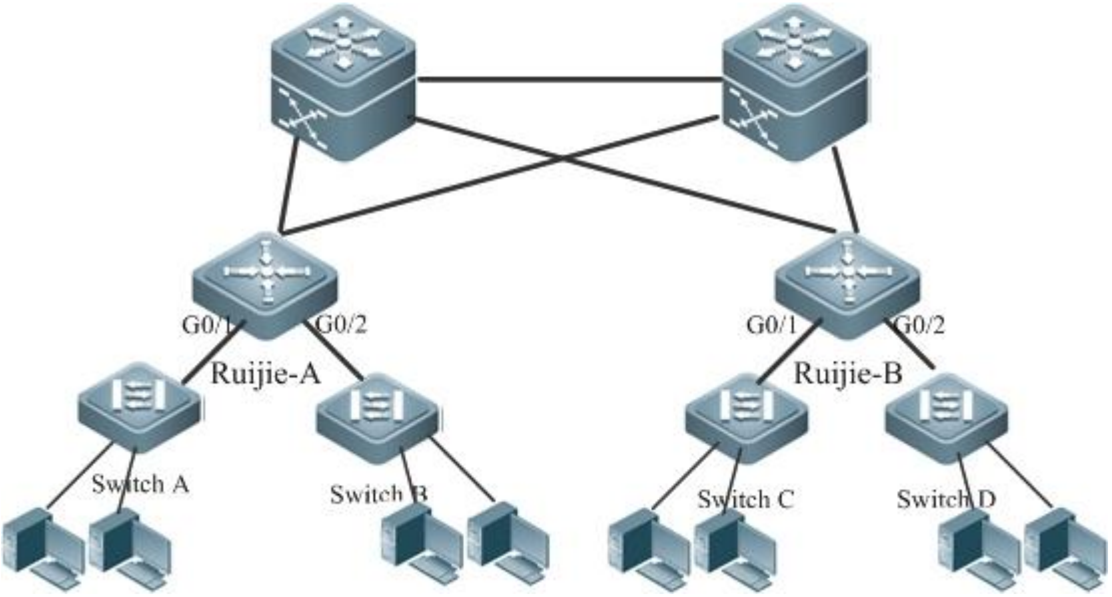


Figure 5 Application of URPF strict mode

Configuration Steps

As shown in Figure 5, enable URPF strict mode on the aggregation-layer device. That is, enable URPF strict mode on Ruijie-A and Ruijie-B.

# Configurations of device Ruijie-A:

```
Ruijie# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# interface gigabitEthernet0/1
Ruijie(config-if)# ip address 195.52.1.1 255.255.255.0
Ruijie(config-if)# ip verify unicast source reachable-via rx
Ruijie(config-if)# ip verify urpf drop-rate notify
```

The configurations to enable URPF strict mode are the same on interface G0/2 of Ruijie-A and G0/1 and G0/2 of Ruijie-B.

Verification

# Verify URPF configurations of Ruijie-A.

```
Ruijie #show ip urpf interface gigabitEthernet 0/1
IP verify source reachable-via RX
IP verify URPF drop-rate notify enabled
IP verify URPF notification threshold is 1000pps
Number of drop packets in this interface is 124
Number of drop-rate notification counts in this interface is 0
```

Example of Loose Mode Configuration

Networking Requirements

The section describes the common application scenario of URPF loose mode, including asymmetrical routing environment and multi-homed network environment.

This section describes the configurations of outlet device connecting the ISP in the multi-homed network as shown in Figure 3.

Networking Topology

Figure 3 shows the multi-homed network.

Configuration Steps

As shown in Figure 3, URPF loose mode is enabled on G3/1 and G3/2 connecting two ISPs to prevent invalid packets from attacking the interior user network and isolate invalid packets outside the user network on the outlet device Ruijie-A in the user network.

# Configurations of Ruijie-A:

```
Ruijie# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# interface gigabitEthernet3/1
Ruijie(config-if)# ip address 195.52.1.2 255.255.255.252
Ruijie(config-if)# ip verify unicast source reachable-via any
Ruijie(config-if)# ip verify urpf drop-rate notify
Ruijie(config-if)# exit
Ruijie(config)# interface gigabitEthernet3/2
Ruijie(config-if)# ip address 152.95.1.2 255.255.255.252
Ruijie(config-if)# ip verify unicast source reachable-via any
Ruijie(config-if)# ip verify urpf drop-rate notify
Ruijie(config-if)# end
```

Verification

# Verify URPF configurations of Ruijie-A.

```
Ruijie #show ip urpf
IP verify URPF drop-rate compute interval is 300s
IP verify URPF drop-rate notify hold-down is 300s
Interface gigabitEthernet3/1
IP verify source reachable-via ANY
IP verify URPF drop-rate notify enabled
IP verify URPF notification threshold is 1000pps
Number of drop packets in this interface is 4121
Number of drop-rate notification counts in this interface is 2
Interface gigabitEthernet3/2
IP verify source reachable-via ANY
IP verify URPF drop-rate notify enabled
IP verify URPF notification threshold is 1000pps
Number of drop packets in this interface is 352
Number of drop-rate notification counts in this interface is 0
```

Configuring IPFIX

Overview

IP Flow Information eXport (IPFIX), is a standard protocol published by the Internet Engineering Task Force (IETF) for the netflow measurement. It standardizes the format of the network traffic statistics. IPFIX is applicable to network devices and management system platforms of any manufacturers, and is used to export the flow statistics by network device. On one hand, IPFIX allows an administrator to easily extract and view the important flow information stored in the network device. On the other hand, it is unnecessary for the administrator to upgrade the network device software or management tools if the flow monitoring requirement changes as the export format is extensible.

IP flow information is transmitted from an export device (router, switch, or network sniffer device) to the collector. Different data formats are defined to meet different requirements, because IPFIX is a highly-scalable template-based format for the data export.

To export the complete data, seven key fields are adopted to represent each network flow: source IP address, destination IP address, source port, destination port, Layer-3 protocol type, byte of Type-of-service, and input logical interface. If all those seven key fields of different IP packets are matched, then all the IP packets belong to the same flow. Network optimization, security detection and traffic accounting can be performed according to the current network application information about recorded features of the netflow, such as flow duration, and the average length of the packet in the flow.

Basic Concept

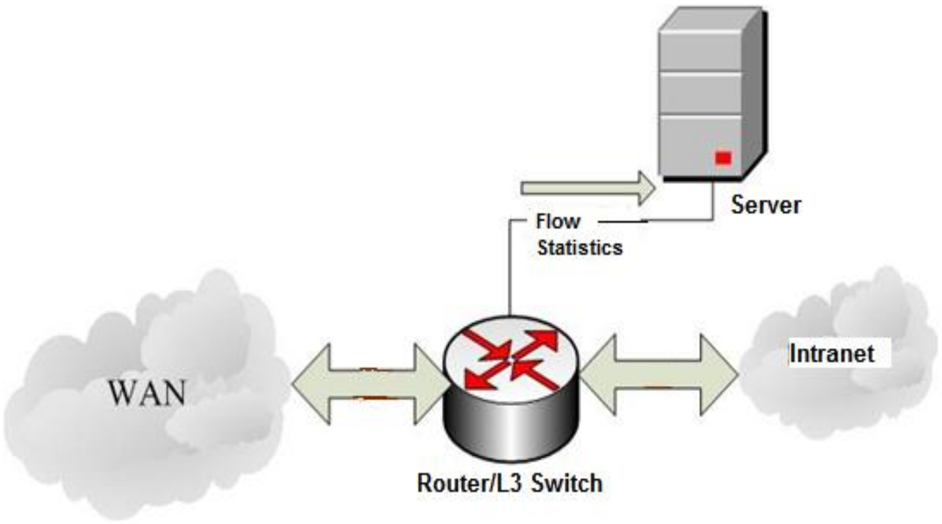
Template: Defines the recorded data format, including a series of data bit domain fields, each of which contains the data type of the data bit domain and the data length. You can parse the recorded data according to the template. If you want to add a data field you are interested in to the data record, you only need to add the corresponding data bit domain field to this template, without modifying the management software.

Collector: Receives the IP flow information generated from the network device. The collector analysis is based on the received data template and data records, and it clearly shows flow information in graphics or tables and saves the information to the database for further use.

IPFIX Application

The router/switch enabled with the IPFIX flow statistics function can collect the statistics on lots of information of packets, including the Layer-3 protocol type, the transport layer port, source/destination address, and service type. The information can be widely used in application scenarios such as user detection, network analysis and planning, security analysis, traffic accounting, and network traffic engineering. Figure-1 shows a typical application topology.

Figure-1



Network Application and User Detection

The IPFIX traffic statistics feature allows you to view detailed, real-time, application-based, and current network usage. It allows you to reasonably allocate and optimize the network resources, and provides the capability of real-time detection of the network capacity. With the IPFIX flow detection function, you can easily understand the network usage and plan to limit the usage combined with other functions such as the Access Control List (ACL). IPFIX can also help effectively and quickly solve some potential security problems.

Network Planning

The IPFIX flow statistics function can detect netflow information over a long period and track network trends. The data enables you to predict network changes and optimize upgrade plans of various networks effectively. It minimizes network costs, but maximizes network performance, capacity and stability. IPFIX can also detect unwanted traffic in Wide Area Network (WAN) and redundant bandwidth and quality of service (QoS) usage. The IPFIX flow statistics function offers valuable information for reducing the cost of operating the network. For example, when the traffic over a WAN link increases, generally you will increase the investment to upgrade the link. However, the traffic increase is possibly attributed to some illegal usage such as BT download. IPFIX enables you to find out the real reason, modify the network usage policy and solve the problem, and thus preventing unnecessary network upgrade.

Security Analysis and Attack Detection

You can use the export flow record of the IPFIX flow statistics to identify denial of service (DoS) attacks, viruses and worms in real-time. Anomalistic changes in network behaviors are clearly reflected in flow records. Take DoS attack for example, its feature is to send a bulk of IP packets (different from the ordinary ones) in the network from untrusted source addresses to the same destination address. Combined with other network control methods (ACL and QoS), the IPFIX flow statistics function can prevent malicious network attack effectively by collecting the source address, destination address, protocol number, port number and size of these packets and sending the information to the collector for network security experts or software analysis.

Flow Accounting

The IPFIX flow statistics function measures the flows in a network in a fine granularity way, including the source/destination IP address, number of packets, total bytes, timestamp, QoS and application ports. Internet Service Providers (ISPs) can utilize the information for accounting based on time, bandwidth, application or network service quality.

IPFIX Function

Note Ruijie IPFIX function is implemented based on the multi-service card only.

Understanding the IP Packet Flow

A packet flow is a series of consecutive packets of the same attributes and pass a same detection point in a period. The packets belong to a same flow have some same attributes, which can be some fields in the head of the IP packets, such as source IP address, Tos field and any combination of those fields. The key fields defining the packet flow are not fixed. For Ruijie switches, a netflow is defined as the unidirectional packet flow with the same source and destination. To be precise, a netflow is determined by the combination of the following key fields:

- Source IP address
- Destination IP address
- Source port number
- Destination port number
- Layer-3 protocol type
- Tos
- Input logical interface
- VRF

Those eight key fields determine a unique flow. If one key field of a packet is different from that of another packet, they belong to different flows. A flow record also includes other statistical fields, such as the packet number, the next-hop IP address, total flow byte number, etc. For each probe packet, ascertain the flow records first, and record the corresponding information stored in the main cache. The flow record cache is used to store the IP flow information, including the main cache and the flow aggregation cache.

Understanding the Main Cache

The main cache is used to store the raw IP flow information, using seven keywords to match the flow statistics. The IPFIX export mechanism records the flow information and sends it to the configured collector. For an IP packet, first search for the corresponding flow record from its keyword in the main cache: if no flow record exists, create a new one; and if the flow record exists, then update the flow statistical information, such as the packet number, flow bytes, etc. The cache capacity is limited; therefore, the aging mechanism of the flow record is set. The following conditions are used to determine whether a flow is aged out or not:

If no packets belong to the flow are detected in a certain period of time (the non-active time), this flow has been aged and shall be exported.

The flow information cannot be recored indefinitely for it persists for a too long time, which exceeds the set active time, then this flow shall be aged.

When it needs to export the aged flow record, the flow information shall be encapsulated as User Datagram Protocol (UDP) packets and sent to the set collector on the server for processing. The collector is the software tool for processing the flow record, and can display the visualized current network status and analyze the netflow according to the received flow record information.

Understanding the Format of the Flow Export Packet

The format of the flow record export packet is the IPFIX standard format. The IPFIX standard format is based on the template, and easy to extend. When encapsulating the data record, first create the format template of the data record, which defines the filed type and length of the data record. For each data record, use the template to identify and explain the analytical format. To add a new field to the date record, you only need to re-create the corresponding data template rather than to upgrade the software, which greatly improves the scalability.

On Ruijie routers, IPFIX supports the exporting of netflow packet formats of version 9 and version 10, and supports the netflow software versions ManageEngine_NetFlowAnalyzer_7001 and ManageEngine_NetFlowAnalyzer_8000. However, the netflow software version ManageEngine_NetFlowAnalyzer_9100 may be problematic on Ruijie routers, because the deployed template is different from Cisco's template.

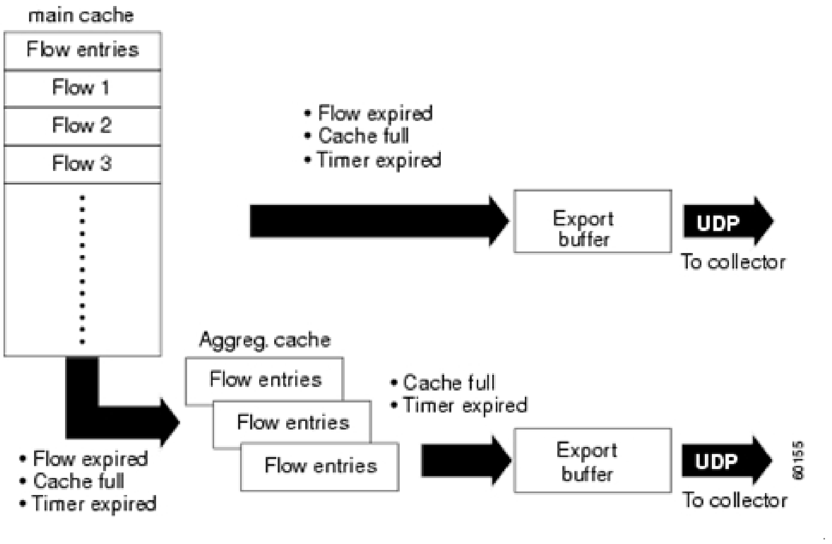
Understanding Flow Aggregation Mode

Similar to main mode, flow aggregation mode also deals with the netflow statistics. The only difference is that the main mode acquires the original packet to generate and export the flow record, while flow aggregation mode reprocesses the flow record to generate and export the new record of the flow aggregation.

Such processing meets requirements of using different key fields to reprocess flow records and generate the required flow records in different flow aggregation modes. As described in the preceding sections, a flow consists of packets of the same attributes, and the attributes can be the combination of any fields in packet headers. In the software, seven keywords differentiate flows, and you can select any seven fields in packet headers to reprocess flow records. For example, to know Layer-3 packet distribution, you can use Layer-3 protocol ID as the key field to aggregate flow records exported in main mode and obtain the required flow information.

Similar to the main mode, flow aggregation mode requires the cache to store the current statistical flow information. The flow aggregation function maps the export flow record in the main cache to the corresponding flow record in the flow aggregation cache, updates the flow aggregation record and regularly checks whether the flow aggregation record expires. Once expired, the flow aggregation record must be exported. Figure-2 shows the principle..

Figure-2:



Configuring the Flow Record Export in Main Mode

When a flow record in the main cache expires, the software uses the export mechanism to encapsulate the expired flow record to a UDP packet and sends the UDP packet to the configured server. Meanwhile, to prevent the loss of the template carrying the flow record because of unrelizable transmission over UDP, it requires for the regular template retransmission to ensure the successful receiving of the data template.

Configuring the Export Destination IP Address and Port

Up to two export servers can be set at the same time to improve the reliability of flow information transmission.

Use the following commands to configure the export server.

Command	Function
Ruijie> **enable**	Enters privileged command mode.
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **ip flow-export destination** *ip-address udp-port* [**vrf** [*vrf-name*]]	Configures the destination IP address and destination port of the flow export, and VRF.
Re-execute the third command	Configures the destination IP addresses and destination ports of multiple flow exports.
Ruijie(config)# **end**	Returns to privileged command mode.
Ruijie# copy running-config startup-config	Saves the configuration.

To remove the flow export, use the **no ip flow-export destination** *ip-address udp-por*t [**vrf** [*vrf-name*]] command.

Configuration example

```
Ruijie# config terminal
Ruijie(config)#ip flow-export destination 192.168.217.76 1111
Ruijie(config)#ip flow-export destination 192.168.217.76 2222
%Warning: Second destination address is the same as previous address 192.168.217.76
Ruijie(config)#ip flow-export destination 192.168.217.76 3333
%Exceeded maximum export destinations
```

```
Ruijie(config)# end
Ruijie products display error information when the third export destination is configured.
```

Caution By default, no flow export is set in the system. The flow record will be saved in the device for checking instead of being exported.

Configuring the Export Source IP Address

A device can have multiple IP addresses, so the IP address on one port can be specified as the source IP address for the sent packet when exporting the flow record.

Use the following commands to configure the source IP address.

Command	Function
Ruijie> **enable**	Enters privileged command mode.
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **ip flow-export source** *interface-number*	Configures the export source IP address as the IP address for an interface.
Ruijie(config)# **end**	Returns to privileged command mode.
Ruijie# copy running-config startup-config	Saves the configuration.

Configuration example

```
Ruijie# config terminal
Ruijie(config)# ip flow-export source gigabitEthernet 6/2
Ruijie(config)# end
Ruijie#
```

The example uses the IP address of g 6/2 as the source address.

Caution By default, the default system IP address is used as the source IP address when sending the packets in the system.

Caution An IP address must have been configured for the port designated as the source address.

Configuring the Related Parameters of the Export Template

Before exporting the netflow data, the corresponding data template shall be sent to the server. UDP, an unreliable protocol, may result in the template loss and cause that the server fails to analyze the flow data correctly. Therefore, the retransmission mechanism is adopted to send the template. There are two RGOS retransmission mechanisms: 1. In the unit of packets, retransmit the template once each time sending *n* data packets; 2. In the unit of minutes, retransmit the template at every certain interval.

Use the following configuration commands.

Command	Function
Ruijie> **enable**	Enters privileged command mode.
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **ip flow-export template refresh-rate** *packets*	Configures the retransmission packet number. The range is from 1 to 600. The default value is 20.
Ruijie(config)# **ip flow-export template timeout-rate** *minutes*	Configures the retransmission interval, in minutes. The range is from 1 to 1000 minutes. The default value is 30 minutes.
Ruijie(config)# **end**	Returns to privileged command mode.
Ruijie(config)# **copy running-config startup-config**	Saves configurations.

Configuration example

```
Ruijie# config terminal
Ruijie(config)#ip flow-export template refresh-rate 30
Ruijie(config)#ip flow-export template timeout-rate 40
Ruijie(config)# end
Ruijie#
```

In the example, *refresh-rate 30* indicates that the template is retransmitted every 30 packets, and *timeout-rate 40* indicates that the template is retransmitted every 40 minutes.

Caution By default, the retransmission packet refresh rate is 20, the time interval is 10 minutes.

Showing the Export Configurations in Main Mode

Use the **show ip flow export** command to display the current export configurations, including the export enable, export destination, the format of the flow record export packet, etc.

Configuring the Main Cache

The main cache is used to save the raw flow record information, and each flow entry size is fixed. A flow entry is created for each active flow in the system, with the record of flow characteristic and statistical information. All flow entries will be regularly checked for determining whether they have expired based on the following conditions:

1. The cache is full and no available space for the flow entries, so some entries shall expire.

2. A flow is inactive. By default, if a flow is not updated within 15 seconds, it becomes inactive.

3. A flow keeps active for too long time. By default, a flow shall expire if it has been active for 30 minutes.

The following introduces those configurable parameters.

Configuring the Flow Cache Entry Number

With the IPFIX flow statistics function enabled on a port, certain storage space is reserved for saving the flow entry in the main cache, which meets the user network demands generally. By default, 64K flow entries (with each entry size 64 bytes) are reserved, so it needs 4M memory space for the main cache. You can increase or decrease the entry number according to your needs to improve the performance or reduce the memory usage, which depends on your device memory size. Use the **ip flow-cache entries** *number* command to set the entry number in the cache.The range of *number* is from 1024 to 524288. The detailed configuration is described as follows.

Use the following configuration commands.

Command	Function
Ruijie> **enable**	Enters privileged command mode.
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **ip flow-cache entries** *number*	Configures the entry number in the cache. The range is from 1024 to 524288.
Ruijie(config)# **end**	Returns to privileged command mode.
Ruijie# **copy running-config startup-config**	Saves the configuration.

To restore the entry number to the default value, use the **no ip flow-cache entries** command.

Configuration example

```
Ruijie# config terminal
Ruijie(config)#ip flow-cache entries 32768
Ruijie(config)# end
```

Caution

By default, 65536 entries are in the main cache. With IPFIX enabled, the entry number configuration in the cache will take effect only after you re-enabling the IPFIX function. It is not recommended to change the entry number in the cache casually and inappropriately, which may result in abnormal system working.

Configuring the Flow Cache Timeout

Use the following command to set the flow cache timeout parameters.

Command	Function
Ruijie> **enable**	Enters privileged command mode.
Ruijie# **config terminal**	Enters global configuration mode.

Command	Function
Ruijie(config)# **ip flow-cache timeout active** *minutes*	Configures the active aging time in minutes. The range is from 1 to 60 minutes.
Ruijie(config)# **ip flow-cache timeout inactive** *seconds*	Configures the inactive aging time in seconds. The range is from 10 to 600 seconds.
Ruijie(config)# **end**	Returns to privileged command mode.
Ruijie# **copy running-config startup-config**	Saves the configuration.

By default, the active aging time is 30 minutes and the inactive aging time is 15 seconds. To restore the default value, use the **no ip flow-cache timeout active** command and the **no ip flow-cache timeout inactive** command.

Configuration example

```
Ruijie# config terminal
Ruijie(config)#ip flow-cache entries 32768
Ruijie(config)#ip flow-cache timeout active 20
Ruijie(config)#ip flow-cache timeout inactive 20
Ruijie(config)# end
```

Showing Information About the Main Cache

Use the **show ip flow cache** command to show the packet flow statistical information in the current main mode, including the packet size distribution, the cache entry usage, etc.

The output is as follows:

```
Ruijie# sh ip flow cache
Ipfix collect data from CM-CARD
ip flow switching cache, 250000 entries
   10 active, 249990 inactive
   active flows timeout in 1 minutes
   inactive flows timeout in 15 seconds
Protocol      Total Flows   Total packets  Total bytes   Active time
udp-other     3            13           834          151
ospf         12           72           5284         622
gre          2            76           1900         114
udp          2            13           834          151
Total:        17          161          8018         887
Display entries in main cache :
SrcIf             SrcIPAddress     DstIf            DstIPAddress     Pr   Tos  SrcPort
DstPort Pkts       ActiveTime  Vrf
Vi1              111.1.1.200      Null0            224.0.0.5       89   0    0      0
0         16        0
Ipfix collect data from device 3
ip flow switching cache, 250000 entries
   17 active, 249983 inactive
   active flows timeout in 1 minutes
   inactive flows timeout in 15 seconds
Protocol      Total Flows    Total packets  Total bytes    Active time
udp-other     8            144          10436         468
ospf         14           82           6144          791
gre          4            253          16182         246
udp          8            144          10436         468
Total:        26           479          32762         1505
Display entries in main cache :
SrcIf            SrcIPAddress     DstIf            DstIPAddress     Pr   Tos  SrcPort
DstPort Pkts       ActiveTime  Vrf
Lo1              22.2.2.2        Gi1/0/1          55.1.1.55        17   0    32768   99
2         12        0
Lo0              20.1.1.1        Gi1/0/1          20.1.1.2         17   0    10000   10000
1         12        0
```

Enabling Flow Statistics

The preceding configurations are performed when the IPFIX flow statistics function is disabled. This section describes how to enable the IPFIX flow statistics function which is to measure the ingress or egress packets. Therefore, to enable the IPFIX flow statics function, you need to set the observation point, flow types, and ports where flows are measured. To enable the IPFIX flow statistics function for ingress or egress flows, you need to enter interface configuration mode.

Use the following configuration commands.

Command	**Function**
Ruijie> **enable**	Enters privileged command mode.
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **interface** *interface-type interface-number*	Enters interface configuration mode.
Ruijie(config)# **ip flow { ingress | egress }**	Enables the IPFIX flow statistics function on the interface.
Ruijie(config)# **end**	Returns to privileged command mode.
Ruijie# **copy running-config startup-config**	Saves the configuration.

To disable the IPFIX flow statistics function, use the **no ip flow { ingress | egress }** command interface configuration modein interface configuration mode.

Configuration example

```
Ruijie# config terminal
Ruijie(config)# interface gigabitEthernet 6/2
Ruijie(config-if)# ip flow ingress          //Enable IPFIX on port 6/2 for statistics of ingress
flows.
Ruijie(config-if)# exit
Ruijie(config)# interface gigabitEthernet 6/3
Ruijie(config-if)# ip flow ingress          //Enable IPFIX on port 6/3 for statistics of ingress
flows.
Ruijie(config-if)# exit
Ruijie(config)# interface gigabitEthernet 6/4
Ruijie(config-if)# ip flow ingress          //Enable IPFIX on port 6/4 for statistics of ingress
flows.
Ruijie(config-if)# exit
Ruijie(config)# interface gigabitEthernet 6/5
Ruijie(config-if)# ip flow ingress          //Enable IPFIX on port 6/5 for statistics of ingress
flows.
Ruijie(config-if)# exit
Ruijie(config)#
```

Use the **show ip flow interface** command to show the IPFIX state on the interface.

Configuration example

```
Ruijie# show ip flow interface
interface gigabitEthernet 6/2
Ip flow ingress
interface gigabitEthernet 6/3
Ip flow ingress
interface gigabitEthernet 6/4
Ip flow ingress
interface gigabitEthernet 6/5
Ip flow ingress
```

Caution

By default, IPFIX is disabled on all interfaces. Once the IPFIX flow statistics function is enabled on an interface, the global IPFIX function is enabled, and the corresponding cache and timer are created. The settings of the main cache will take effect when the IPFIX function is re-enabled. Use the **no** form to disable all interface with IPFIX enabled, and then the global IPFIX takes no effect.

The IPFIX captured data flow is sent to the data flow analysis software (Such as NewFlow,RILL) and then be converted into a visible report for the user. Because the IPFIX flow statistics function is enabled on an interface.When the system restarted, the interface index might change and the IPFIX is unable to capture the data flow on the original designated interface. To avoid this consequence, use the command snmp-server if-index persist to bind the function on a interface.

Configuring Flow Aggregation Mode

Flow Aggregation Mode Overview

Flow aggregation mode re-aggregates the main mode flow and generates a new flow through the defined specified key field. The system reserves certain cache, which is similar to the main cache and here called the flow aggregation cache, for aggregation mode. When a flow entry comes out of the main cache, the flow information is used to refresh the flow aggregation record in each enabled flow aggregation cache. You could set the entry number in the aggregation cache, entry aging parameter, export destination IP address and export destination UDP port separately. Meanwhile, the entry aging mechanism in the flow aggregation mode, which supports the forcible aging according to the user requirements, is the same as the one in main mode. By default, the entry number in the flow aggregation cache is 4096.

The following flow aggregation modes are supported:

- Destination Prefix aggregation mode
- Prefix aggregation mode
- Protocol Port aggregation mode
- Source Prefix aggregation mode
- Destination Prefix-ToS aggregation mode
- Prefix-port aggregation mode
- Prefix-ToS aggregation mode
- Protocol-port-ToS aggregation mode
- Source Prefix-ToS aggregation mode

Aggregation Mode	Key Field
destination-prefix	Destination AS number, destination address mask length, destination prefix and egress interface index
prefix	Source AS number, destination AS number, source address mask length, destination address mask length, source prefix, destination prefix and egress interface index.
prefix-port	Source prefix, destination prefix, source port, egress interface index and ToS value.
protocol-port	Protocol number, source port and destination port.
source-prefix	Source AS number, source address mask length, source prefix.
destination-prefix-tos	Destination AS number, destination mask length, destination prefix and egress interface index.
prefix-tos	ToS, source AS number, source prefix, source mask length, destination AS number, destination mask length, destination prefix.
protocol-port-tos	ToS, protocol type, source port and destination port.
source-prefix-tos	ToS, source prefix, source mask length and source interface index.

The nine modes are independent and can be configured concurrently.

Configuring Flow Aggregation Mode

IPv4 Aggregation Mode

Configuration example

```
enable
configure terminal
```

```
ip flow-aggregation cache { destination-prefix | destination-prefix-tos | prefix  | prefix-port
| prefix-tos | protocol-port | protocol-port-tos | source-prefix | source-prefix-tos}
cache entries number
cache timeout active minutes
cache timeout inactive seconds
export destination ip-address udp-port
Repeat the step7, set the second export destination.
enabled
exit
interface interface-type interface-number
ip flow {ingress | egress}
exit
Repeat the step11-13, set the IPFIX function on other interfaces.
end
```

The following table describes the configuration commands in detail.

Command	Function
Ruijie> **enable**	Enters user execution mode.
Ruijie# config terminal	Enters global configuration mode.
Ruijie(config)# ip flow-aggregation cache { destination-prefix | destination-prefix-tos | prefix | **prefix-port | prefix-tos** | **protocol-port** | **protocol-port-tos** | **source-prefix** | **source-prefix-tos** }	Enters corresponding flow aggregation mode. destination-prefix: enters destination-prefix aggregation configuration mode. destination-prefix-tos: enters destination-prefix-tos aggregation configuration mode. prefix: enters prefix aggregation configuration mode. prefix-port: enters prefix-port aggregation configuration mode. prefix-tos: enters prefix-tos aggregation configuration mode. protocol-port: enters protocol-port aggregation configuration mode. protocol-port-tos: enters protocol-port-tos aggregation configuration mode. source-prefix: enters source-prefix aggregation configuration mode. source-prefix-tos: enters source-prefix-tos aggregation configuration mode.
Ruijie(config-flow-cache)# **cache entries number**	Configures the cache entry number. number: indicates the allowed cache entry number in this aggregation mode. The range is from 1024 to 524288. The default value is 4096.

Command	Function
Ruijie(config-flow-cache)# **cache timeout active minutes**	(Optional) Configures the cache entry active timeout time. minutes: indicates the active timeout time. The range is from 1 to 60 minutes. The default value is 30 minutes.
Ruijie(config-flow-cache)# **cache timeout inactive seconds**	(Optional) Configures the cache entry inactive timeout time. seconds: indicates the inactive timeout time. The range is from 10 to 600 seconds. The default value is 15 seconds.
Ruijie(config-flow-cache)# **export destination ip-address udp-port**	(Optional) Configures the flow aggregation export destination. ip-address: indicates the export destination IP address. udp-port: indicates the destination UDP port number.
Repeat the previous step, set the second export destination.	(Optional) Configures up to two export destination for each aggregation mode.
Ruijie(config-flow-cache)# **enabled**	
Ruijie(config-flow-cache)# exit	Exits flow aggregation configuration mode and enters global configuration mode.
Ruijie(config)# **interface ethernet** 0/0	Enters interface configuration mode.
Ruijie(config-if)# ip flowegress	Enables the IPFIX function on the interface. Ingress: detects the ingress flow to the interface; Egress: detects the egress flow from the interface.
Ruijie(config-if)# **end**	Exits interface configuration mode.

Configuration example

```
Ruijie> enable
Ruijie# configure terminal
Ruijie(config)# ip flow-aggregation cache destination-prefix
Ruijie(config-flow-cache)# cache entries 2048
Ruijie(config-flow-cache)# cache timeout active 15
Ruijie(config-flow-cache)# cache timeout inactive 300
Ruijie(config-flow-cache)# export destination 172.30.0.1 991
Ruijie(config-flow-cache)# enabled
Ruijie(config-flow-cache)# exit
Ruijie(config)# interface ethernet 0/0
Ruijie(config-if)# ip flow egress
Ruijie(config-if)# end
```

Showing Flow Aggregation Information

IPv4 configuration commands

show ip flow cache aggregation { **as** | **as-tos** | **destination-prefix** | **destination-prefix-tos** | **prefix** | **prefix-port** | **prefix-tos** | **protocol-port** | **protocol-port-tos** | **source-prefix** | **source-prefix-tos** } [vrf *vrf-name*]

Use this command to show the cache information in each flow aggregation mode, including the cache size, effective flow entry number, idle flow entry number, etc.

Configuration example

```
Ruijie# sh ip flow cache aggregation protocol-port
IP Flow Switching Cache, 278544 bytes
2 active, 4094 inactive, 12523 added
239947 ager polls, 0 flow alloc failures
Active flows timeout in 30 minutes
Inactive flows timeout in 15 seconds
IP Sub Flow Cache, 17160 bytes
0 active, 1024 inactive, 0 added, 0 added to flow
0 alloc failures, 0 force free
1 chunk, 1 chunk added
Protocol  Source Port  Dest Port  Flows  Packets  Bytes/Packet  Active
0x11      0x007B     0x007B     1       1       76
0.0
0x01      0x0000     0x0303     1       5      132
5.0
Ruijie#
```

show ip flow export

Use this command to show the export information in main mode, and the export configuration information about enabled aggregation modes, such as the format of the ouput packet, export destination, etc.

Configuration example

```
Ruijie# sh ip flow export
Ipfix collect data from CM-CARD
cache for main metering process:
       flow export is enabled
       Exporting flows to 55.1.1.55 (99)
       Exporting using source interface Loopback 1
       Template export information:
         Template timeout = 10 minutes
         Template refresh rate = 20 packets
       total 400 packets metering
       total 0 packets dropped for no memory
       total 42 flows exported in 5 udp datagrams
       0 ipfix message export failed

Ipfix export information from device 3
cache for main metering process:
       flow export is enabled
       Exporting flows to 55.1.1.55 (99)
       Exporting using source interface Loopback 1
```

```
Template export information:
  Template timeout = 10 minutes
  Template refresh rate = 20 packets
total 1124 packets metering
total 0 packets dropped for no memory
total 62 flows exported in 9 udp datagrams
0 ipfix message export failed
```

Configuring the Flow Filtering and Sampling Mechanism

With the netflow growth, concurrent netflows set in, affecting the performance of the IPFIX flow measurement. Then the flow filtering and sampling mechanism comes into being. The flow filtering analyzes the flows which the users are interested in, decreasing the IPFIX flow and network device loads sharply. The sampling mechanism analyzes some netflows randomly according to a certain sampling rate, decreasing the IPFIX flow greatly.

Configuring Sampling for the Specified Flow

The sampling for the specified flow uses the ACL to match the packets, and only the matched packet flow is recorded.

Use the following configuration commands.

Command	Function
Ruijie> **enable**	Enters privileged command mode.
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **interface** *interface-name*	Enters interface configuration mode.
Ruijie(config-if)# **flow-sample** *packet-num* filter *acl-name*	Applies the ACL to the interface to filter the input flow. *acl-name* is the existed ACL ID or name, it can also be 0, which means all flows are permitted. The matched packets are sampled by the ratio of 1/*packet-num*, and then flow statistics are collected.
Ruijie(config)# **end**	Returns to privileged command mode.
Ruijie# **copy running-config startup-config**	Saves the configuration.

To restore the default configuration of an interface, use the **no** form of this command in interface configuration mode.

All packets of ports are sampled and measured by the ratio 1/255 by default.

⚠ Caution

If the IPFIX flow measurement is disabled on the configured interface, this configuration will be saved. Once IPFIX is enabled on the port, this configuration takes effect. The corresponding ACL must exist when you configure this command. If the ACL is deleted, this configuration will be deleted automatically.

For example:

```
Ruijie# config terminal
Ruijie(config)# interface gi 2/2
Ruijie(config-if)# flow-sample 100 filter acl1
Ruijie(config-if)# end
```

Configuring RLOG

Overview

The device side is responsible for log collection and uploading, while RLOG will send all Internet access information and user connection information to the server. The server-side software will analyze the logs and then write the logs into the database, and the user can then find the corresponding connection records through log query system.

RLOG contains NAT logs and the number of bytes received/sent, as well as connection establishment/deletion and other relevant information.

To enable logging function, you need to complete the following configurations:

- Configure logging service on device side and enable flow log;
- Enable background service program on the service side;
- Configure web server.

Log configuration

Log Server Configuration

Configure log server to enable logging function. If no log server is configured, the device will not send any log to the log server.

When the log server is configured, the device will enable the logging module and send out log information in UDP packets.

To configure log server, execute the following commands in global configuration mode:

Command	Function
Ruijie(config)# **rlog server** *server-ip* [**vrf** *vrf-name*]	Specify the IP address and VRF name of log server and enable log service;
Ruijie(config)# **no rlog server**	Remove log server configuration, disable logging service and clear relevant statistics.

Note

The command to configure log server will only enable log service, and the log output function is not enabled at the same time. Executing this command along will not output any log. The command to enable flow log is "**ip session log-on**", which must be executed separately.

Log Service Parameter Configuration

Log service related parameters include the maximum length of log packets, number of service port.

By executing these commands, the user can modify the configurations of log server and avoid the conflict between log service and other network services.

To configure log service parameters, execute the following commands in global configuration mode:

Command	Function
Ruijie(config)# **rlog mtu** *number* Ruijie(config)# **no rlog mtu**	Configure maximum length of log packets Remove the configuration of maximum length of log packets and restore to the default 1500
Ruijie(config)# **rlog port** *number* Ruijie(config)# **no rlog port**	Specify the log service port number Remove the configuration of log service port number and restore to the default 10000.
Ruijie(config)# **rlog export-rate** *number* Ruijie(config)# **no rlog export-rate**	Specify the log service export rate (maximum number of logs sent per second) Remove the configuration of log service export rate and restore to the default 1000

Note

Any change to the log service parameters will not take effect immediately. You need to restart log service to apply the configurations. You can reconfigure log server to reboot the server.

Note

The default value for log export rate is on the low side. You can configure to the maximum value if the performance of log server allows.

Log Service Testing

You can execute log service testing command to check whether the logging function is normal. This command is used to check free buffer and send test packets to the log server. If the log server receives the test packets and replies with the corresponding prompting messages, we can then determine that whether the log service is properly configured and whether the network is accessible.

To test the log server, execute the following command in global configuration mode:

Command	Function
Ruijie(config)# **rlog test**	Test log server and check free buffer; send test packets to the log server.

Note

Checking free buffer may occupy full log buffer and result in the loss of logs. Please don't use this function unless it is necessary.

Log Service Statistics

Log service statistics also includes the current configuration information, the number of logs received, the number of logs sent, the number of errors sent, and the cause of the latest error.

To display log service statistics, execute the following command in global configuration mode:

Command	Function
Ruijie(config)# **show rlog**	Display log service statistics.

Log Database

The log server supports MySQL, SQLSever, Oracle and etc.

Log data are long-term and continuous records. The storage of long-term data will require sufficient storage space. Different on-site service conditions will have different requirements on database.

The following is a simple example for explaining the size of database required.

Approximate 4,000 users in one school access Internet through router. Assuming that there are averagely 8k new connections and each connection contains 50 bytes of data.

Data per second: 8k*50 = 400k

Daily data: 400k*60*60*24 = 34.5G

Monthly data: 34.5*30 = about 1T

If there are more users or if the storage time is longer, then the storage space shall be increased accordingly.

Caution

The device will only send logs ceaselessly, while the log server will only analyze the logs and write into the database. The user shall be responsible for data maintenance.

Caution

For example, logs to be kept for only 15 days can be deleted by adding a trigger configured to delete data exceeding 15 days at 12:00pm of everyday.

Caution

The logs will keep all connection information, as well as some attack attempts or invalid connections. If these data are considered unnecessary, you can add a trigger configured to periodically records with 0 received byte in order to optimize the data.

Caution

These actions cannot be achieved by programs on the device or log server.

Web pages are directly related to database. Any change to the database (address, username, password and other configuration items) will need to change the corresponding web page.

Log Server

Please refer to the enclosed CD for content and web configurations of the server.

Configuring WAN-TA

Understanding WAN-TA

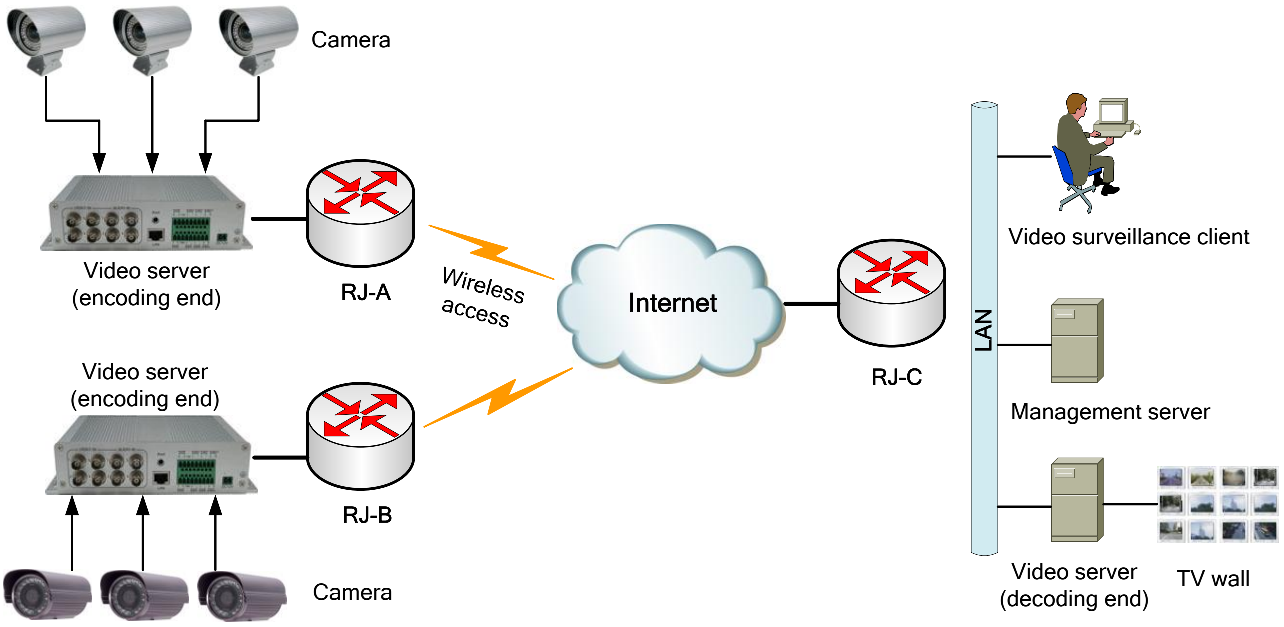
Overview

The video transmission optimization solution is mainly used in network video surveillance scenarios. This document briefly introduces the network video surveillance system and the video transmission optimization solution.

Network Video Surveillance System

The network video surveillance system is generally a real-time remote video surveillance system based on an IP network. It involves the processes of video collection, digital video encoding, data transmission, video decoding, and video playing. A typical video surveillance system consists of front-end surveillance devices (such as a camera, a microphone, and a video server), an IP network, and display devices (video clients). After the camera collects image signals and the microphone collects voice signals, the signals access the video server. The video server performs analog-to-digital conversion, H.264-based video compression and other digital processing, and releases the data to the network. The data reaches the video client after traversing a LAN, a wireless network, and the Internet. Then the video client decodes the data and plays the data on a screen. Figure 0-1 shows the typical structure of the wireless network video surveillance system.

Figure 0-1 Wireless Network Video Video data has two characteristics during transmission:



1) Mass data: A large volume of video data and information is involved.

2) Irregular traffic: The video data volume changes over time.

The characteristics, however, result in many challenges when video data is transmitted on a wireless network. It is well known that the wireless network has poor stability, such as instable bandwidths, high delay, high jitter, and high packet loss rates. The video server at the sending end sends video frames at a fixed interval (a video frame is composed of one or more packets). If video packets are transmitted through an unstable network, the video packets may reach the receiving end after an irregular delay. As a result, video played after being decoded may pause or stutter, or even that the video connection is interrupted and needs to be set up again.

Video Transmission Optimization Solution

The video transmission optimization solution shown in Figure 1-1 is implemented as follows: Firstly, enable WAN-TA on the sending end (RJ-A or RJ-B) to enhance video transmission quality; secondly, enable WAN-TA and RTP shaping on the receiving end (RJ-C) to eliminate video pause and stutter caused by network jitter. This solution enables videos transmitted through networks to be more smoothly played, reduces video pause and stutter, and enhances the video effect.

Basic Concepts

Main Protocols

The network video surveillance system mainly involves the protocols TCP, RTSP, and RTP/RTCP.

The Transmission Control Protocol (TCP) provides a mechanism that allows applications to reliably exchange data on a network. Although TCP provides a connection-oriented and reliable transmission environment, the network where applications are deployed has been greatly changed over the past two decades. In many cases, TCP has become a bottleneck for WAN application performance. The low transmission efficiency of TCP is outstanding on a wireless network (3G/4G), and therefore is urgent to be solved.

The Real Time Streaming Protocol (RTSP) defines a process for one-to-multiple applications to efficiently transmit multimedia data through an IP network, and provides the stop, fast forward, fast backward, and positioning functions for video and audio streams. RTSP is also an application-layer protocol and works with RTP and other lower-layer protocols to provide complete sets of services based on the Internet. Generally, RTSP uses TCP to complete control information transmission and uses RTP to complete media data transmission. The TCP port 554 is used by RTSP.

The Realtime Transport Protocol (RTP) is a protocol for transmitting multimedia data streams on the Internet. RTP is mainly used to transmit real-time streams in network video surveillance applications. It is defined to work in one-to-one or one-to-many transmission scenarios, aiming to provide time information and synchronize streams. RTP only ensures transmission of real-time data, but cannot provide a mechanism for transmitting data packets in order. It does not provide the traffic control or congestion control function, either. RTP provides services based on the Realtime Transport Control Protocol (RTCP).

WAN-TA

WAN Transmission Acceleration (WAN-TA) is a generic name of technologies for enhancing TCP transmission efficiency on WAN links. To enhance the TCP transmission efficiency in a video network transmission environment (in the network video surveillance system, video packets are encapsulated in TCP packets), Ruijie routers have ushered in new TCP features through WAN-TA and applied the features to data streams to be forwarded so as to improve TCP transmission performance on WAN links. WAN-TA splits a TCP connection that is set up through a Ruijie router into two connections, so that the router can serve as a client to participate in TCP sessions, and control TCP data streams through a WAN-TA optimization policy on the router. WAN-TA can eliminate nearly all TCP barriers without changing the client, the server, or network features.

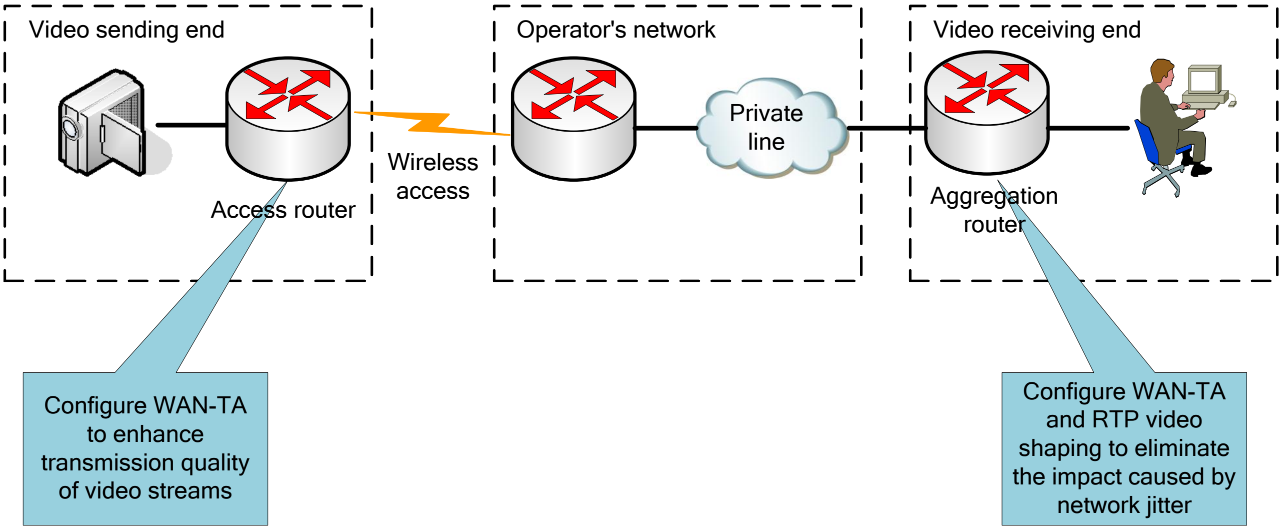
RTP Shaping

RTP shaping mainly uses the delay technology to ensure that RTP service packets reach the peer end at a fixed interval. Based on WAN-TA, RTP shaping fetches packets from a WAN-TA receiving queue, and caches video frames in an RTP queue. After a specific period of time (generally hundreds of milliseconds to several seconds), video frames are sent one after another according to the original time sequence at a fixed interval. This allows the video client to receive stable video streams.

Working Principle

As shown in the wireless network video transmission system in Figure 0-2, the video server is connected with the access router through a wired line, and the access router is connected with the video client through a wireless link. When WAN-TA is configured on the access router, the access router can use the air interface bandwidth to the utmost, thereby accelerating traffic transmission from the server to the client (in the outbound direction). Configured with WAN-TA and RTP shaping, the aggregation router can receive and cache video packets and send them to the video client at a fixed interval, allowing the client to obtain stable video streams. WAN-TA on the receiving router and RTP shaping on the aggregation router work together to ensure the best video effect. If WAN-TA is not configured on the access router to ensure video data transmission, videos played on the client may stutter frequently, or even that the connection is interrupted and needs to be set up again. If RTP shaping is not configured on the aggregation router for caching and shaping video packets, videos played on the client may pause frequently.

Figure 0-2 Deployment Mode of the Video Transmission Optimization Solution


Video sending end
Access router
Wireless access
Operator's network
Private line
Video receiving end
Aggregation router
Configure WAN-TA to enhance transmission quality of video streams
Configure WAN-TA and RTP video shaping to eliminate the impact caused by network jitter

Protocol Specification

The video transmission optimization solution involves the following RFC specifications:

- RFC793: TCP
- RFC2326: RTSP
- RFC3550: RTP

Configuring WAN-TA

Default Configuration

Feature	Default Setting
Globally enabling WAN-TA	WAN-TA is disabled by default.
Configuring a WAN-TA policy	No WAN-TA policy is configured.
Configuring the congestion control algorithm of the WAN-TA policy	The default algorithm is low-bandwidth-delay.
Configuring the WAN-TA policy to support SACK	The SACK option of the WAN-TA policy is enabled.
Configuring whether the WAN-TA policy supports the timestamp option	The timestamp option of the WAN-TA policy is enabled.
Configuring whether the WAN-TA policy supports the window scaling option	The window scaling option of the WAN-TA policy is enabled.
Configuring the send queue depth of the WAN-TA policy	The send queue depth is 1024 by default.
Configuring the window compensation of the WAN-TA policy	The window compensation is 0 by default. That is, no window compensation is added.
Configuring the initial window of the WAN-TA policy	The initial window size is 10 MSS by default.
Configuring the MSS of the WAN-TA policy	The MSS is 1420 by default.
Configuring the keepalive packet sending interval and the maximum number of retries of the WAN-TA policy	The keepalive packet sending interval is 120 minutes, and the maximum number of retries is 9 by default.
Applying the WAN-TA policy to an interface	WAN-TA is disabled on an interface.
Configuring the matching port of the WAN-TA policy	The default matching port number is 554.

Configuration Guide

- The inbound interface and the outbound interface of data flows to which WAN-TA is applied must be express-forwarding interfaces.
- All TCP data flows optimized by WAN-TA are forwarded by the device running WAN-TA.

Globally Enabling WAN-TA

Command	Function
Ruijie(config)# **wan-ta enable**	Globally enables the WAN-TA function. The WAN-TA function takes effect on an interface only when the WAN-TA function is globally enabled and a WAN-TA policy is applied to the interface.

To disable the WAN-TA function, use the **no wan-ta enable** command in global configuration mode.

Configuration example:

The following example enables the global WAN-TA function:

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
```

```
Ruijie(config)# wan-ta enable
```

Note

After the WAN-TA function is enabled, it takes effect for only new flows instead of existing flows. If you hope that the WAN-TA function takes effect for an existing connection, you need to disconnect and then re-create the connection after enabling the WAN-TA function.

Note

After the global WAN-TA function is disabled, the existing TCP sessions to which WAN-TA is applied will continue to be effective.

Creating a WAN-TA Policy

Command	Function
Ruijie(config)# **wan-ta policy { video |** *policy-name* } [**cong-algorithm** *cong-value* | **init-cwnd** *cwnd-value* | **sack { enable | disable} | mss** *mss-value* | **keepalive interval** ***i****nterval-value* **retry** *retry-count* **]**	Creates a WAN-TA policy and specifies the parameters of the WAN-TA policy. This command can be configured in the following two ways: 3) Directly specify parameters behind the command. Then the command does not enter the following WAN-TA policy configuration mode. 4) Do not specify any parameter behind the command. Then the command enters the following WAN-TA policy configuration mode, and you can specify the parameters of the WAN-TA policy one by one.
Ruijie(config-wan-ta-policy)#**cong-algorithm { default | high-delay-1 | high-delay-2 | high-lost | low-bandwidth-delay }**	Configures the congestion algorithm of the WAN-TA policy. This command has the same function as the cong-algorithm *cong-value* parameter of the wan-ta policy command.
Ruijie(config-wan-ta-policy)# **sack { enable | disable }**	Enables or disables the SACK function of the WAN-TA policy. This command has the same function as the sack parameter of the wan-ta policy command.
Ruijie(config-wan-ta-policy)# **init-cwnd** *cwnd-value*	Configures the initial window size of the WAN-TA policy. The value range is 2 to 10. This command has the same function as the init-cwnd *cwnd-value* parameter of the wan-ta policy command.
Ruijie(config-wan-ta-policy)#**mss** *mss-value*	Configures the MSS of the WAN-TA policy. The value range is 68 to 1460. This command has the same function as the mss *mss-value* parameter of the wan-ta policy command.
Ruijie(config-wan-ta-policy)#**keepalive interval** *interval-value* retry *retry-count*	Configures the keepalive packet sending interval of the WAN-TA policy within the range from 2 to 300 minutes and the maximum number of retries of the keepalive

Command	Function
	packet within the range from 1 to 9. This command has the same function as the keepalive interval *interval-value* retry *retry-count* parameter of the wan-ta policy command.
Ruijie(config-wan-ta-policy)#**peer window-compensation** *cmp-value*	Configures the peer window compensation of the WAN-TA policy. The value range is 0 to 655350.
Ruijie(config-wan-ta-policy)#**queue-send-depth** *dep-value*	Configures the send queue depth of the WAN-TA policy. The value range is 8 to 4096.
Ruijie(config-wan-ta-policy)#**timestamp disable**	Disables the timestamp option of the WAN-TA policy. Use the no form of this command to enable the timestamp option.
Ruijie(config-wan-ta-policy)#**window-scale disable**	Disables the window scaling option of the WAN-TA policy. Use the no form of this command to enable the window scaling option.
Ruijie(config-wan-ta-policy)#**match-port** *port-number [port-number…]*	Configures the matching port of the WAN-TA policy. The port number range is 1 to 65535 (554, by default). Use the **no** form of this command to delete all matching ports.

To delete the configured WAN-TA policy, use the **no wan-ta policy video** command in global configuration mode.

Configuration example 1:

The following example creates a system-defined WAN-TA policy named "video":

```
Ruijie(config)#wan-ta policy video
```

Configuration example 2:

The following example creates a user-defined WAN-TA policy named "ftp" and specifies parameters behind the command:

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)# wan-ta policy ftp init-cwnd 8 cong-algorithm low-bandwidth-delay
```

Configuration example 3:

The following example creates a user-defined WAN-TA policy named "ftp" and enters the policy mode to specify parameters:

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#wan-ta policy ftp
Ruijie(config-wan-ta-policy)#init-cwnd 8
Ruijie(config-wan-ta-policy)#cong-algorithm low-bandwidth-delay
```

Caution It is not recommended that you specify parameters when creating a system-defined policy, such as the policy in configuration example 1. The system automatically generates the parameters. It is not recommended that you modify the settings of the parameters.

Caution When creating a user-defined policy, you can directly use the **wan-ta policy** command with parameters or enter the WAN-TA policy configuration mode to specify parameters one by one. For details, see configuration examples 2 and 3.

Caution In WAN-TA policy configuration mode (**wan-ta-policy**), you can configure the peer window compensation and send queue depth, disable the timestamp option, and disable the window scaling option. However, it is recommended that generally you keep the default settings of these parameters instead of manually configuring them.

Caution Before deleting a WAN-TA policy, you need to remove the WAN-TA policy from interfaces on which the WAN-TA policy has been applied. If you modify the configuration after a policy is applied on an interface, the changes do not take effect for the existing connections to which WAN-TA is applied.

Enabling a WAN-TA Policy on an Interface

Command	Function
Ruijie(config-if-gigaethernet0/0)# **wan-ta-policy video list** { *acl-id* | *acl-name*}	Applies a WAN-TA policy to an interface. The *acl-id* or *acl-name* parameter specifies the ACL that matches the flow to be optimized.

To delete the configured WAN-TA policy from an interface, use the **no wan-ta-policy** { *policy-name* | **video** } command in interface configuration mode.

Configuration example:

The following example applies a WAN-TA policy named "video" to an interface, so that all flows matching ACL 101 will be accelerated according to the WAN-TA policy named "video":

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)#interface gigaethernet0/0
Ruijie(config-if-gigaethernet0/0)# wan-ta-policy video  list 101
```

Caution When there is a TCP connection to which a WAN-TA policy is applied on an interface, the WAN-TA function still takes effect on the interface even after the WAN-TA policy is deleted from the interface.

Monitoring and Maintaining WAN-TA

Command	Function
Ruijie# **show wan-ta policy session**	Shows summary information about TCP sessions to which WAN-TA has been applied.
Ruijie# **show wan-ta policy session** *session_id*	Shows statistics about the session with the specified session ID.
Ruijie# **show wan-ta policy video**	Shows configuration information about a WAN-TA policy.

Configuration Examples

Configuration Example of Video Transmission Optimization

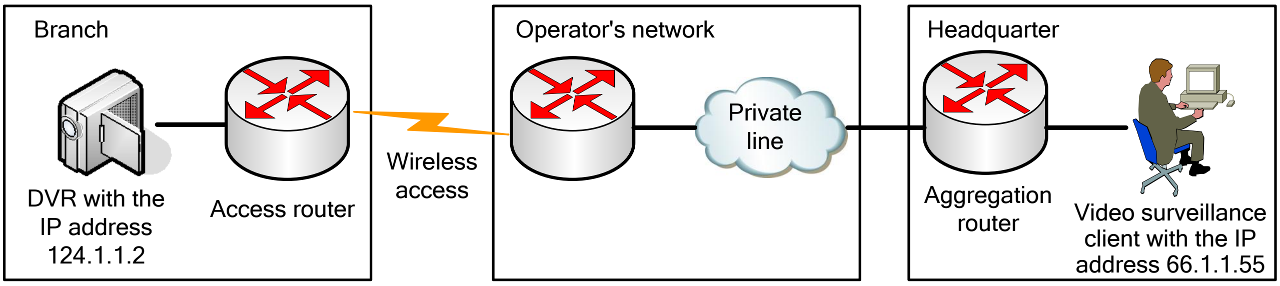
Networking Requirements

As shown in the wireless video surveillance scenario in Figure 0-3, the access router in the branch is connected to the video server (DVR); the aggregation router in the headquarter is connected to the video client.

- The access router is connected to the headquarters through a 3G/4G link.
- The video surveillance connection between the branch and the headquarter is set up based on TCP.
- Before reaching the router to which an optimization policy applies, data streams are not encrypted or encapsulated in other tunnels.

Networking Topology

Figure 0-3 WAN-TA Application Topology



Configuration Tips

On the access router:

- Globally enable WAN-TA.
- Identify the features of data streams to which WAN-TA applies.
- Create a WAN-TA optimization policy.
- Enable the WAN-TA optimization policy on the interfaces that data streams pass through.

On the aggregation router, configure WAN-TA and RTP shaping:

- Identify the features of data streams to which RTP applies.
- Configure RTP shaping.

Configuration Steps

1. Configure and enable WAN-TA on the access router.

5) Configure an ACL for matching data streams that need to be optimized.

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)#ip access-list extended 101
Ruijie(config-ext-nacl)#10 permit ip any any
```

6) Globally enable WAN-TA.

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)#wan-ta enable
```

7) Create a WAN-TA optimization policy. For example, create an optimization policy named "video" to optimize the video surveillance connection between the branch and the headquarter. Video surveillance data is generally encapsulated through RTSP on the TCP port 554.

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# wan-ta policy video
Ruijie(config-wan-ta-policy)#match-port 554 37777
```

8) Enable the optimization policy on a port. You can enable the WAN-TA optimization policy on either the inbound interface gigaethernet0/1 or the outbound port on the access router.

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)#int gigabitEthernet 0/1
Ruijie(config-if-GigabitEthernet 0/1)#wan-ta-policy video list 101
```

2. Configure WAN-TA and enable RTP shaping on the aggregation router.

9) Configure and enable WAN-TA on the aggregation router, because RTP shaping is based on WAN-TA.

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)#ip access-list extended 111
Ruijie(config-ext-nacl)#10 permit ip any any
Ruijie(config-ext-nacl)#exit
Ruijie(config)#wan-ta enable
Ruijie(config)#wan-ta policy video
Ruijie(config-wan-ta-policy)#match-port 554 37777
Ruijie(config-wan-ta-policy)#exit
Ruijie(config)#interface gigabitEthernet 2/1/0
Ruijie(config-if-GigabitEthernet 2/1/1)#wan-ta-policy video list 111
```

10) Configure RTP shaping on either the inbound port or the outbound port on the aggregation router.

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)#
Ruijie(config)#traffic classifier tc1 or
Ruijie(config-traffic-classifier)# if-match acl 111
Ruijie(config-traffic-classifier)#
Ruijie(config-traffic-classifier)#traffic behavior tb1
Ruijie(config-traffic-behavior)# rtp-shaping delay 1000 clock-rate 90000
Ruijie(config)#traffic policy tp1
Ruijie(config-traffic-policy)# classifier tc1 behavior tb1 precedence 1
Ruijie(config-if-GigabitEthernet 2/1/0)#traffic-policy tp1 inbound
```

Verification

11) Check WAN-TA optimization policy configuration.

```
Ruijie#show wan-ta policy video
wan-ta policy: video
   Congestion Control : low-bandwidth-delay
   SACK Support: TRUE
   Initial Congest Window: 10 MSS
   Maxitum Segment Size: 1460
Keepalvie Interval(retry): 120(9)
Match Port: 554 37777

apply on interfaces:
interface name                      list
GigabitEthernet 2/1/0                 101
```

12) Check the current session.

```
Ruijie#sh wan-ta  policy session  vtty 2/1
session_id  pair    flow                          tcp_state        uptime     service
391        392     [124.1.1.2:554->66.1.1.55:1776]    TCP_ESTABLISHED   0:00:06    RTSP
392        391     [66.1.1.55:1776->124.1.1.2:554]    TCP_ESTABLISHED   0:00:06    RTSP
```

The session proxy with the session ID 392 deals with LAN communication with the off-bank ATM end. The session proxy with the session ID 391 deals with WAN communication with the headquarter end. Ruijie focuses on WAN communication.

13) Check the running status of a session on a link.

```
Ruijie#sh wan-ta  policy session vtty 2/1 391


[124.1.1.2:554->66.1.1.55:1776]
```

```
(timer notify: 7898, handle: 2177, signal handle: 4205245876, while: 1702656051)
efb_total_num: 50115, efb_avbuf_num: 48476
sock state: TCP_ESTABLISHED      ref_cnt: 2
Congestion Control:
  algorithm    : low-bandwidth-delay
  icsk_ca_state : Disorder       icsk_retransmits: 11
  icsk_rto     : 6720 ms        icsk_timeout   : 4180 ms
  icsk_backoff  : 5             icsk_probes_out : 0
  disorder_num  : 12            cwr_num       : 0
  recovery_num  : 1             loss_num      : 0
TCP Options:
  is_tstamp    : no
  is_sack      : yes
  is_wscale    : yes        snd_wscale: 0          rcv_wscale: 0
Statistics:
  unacked      : 1             sacked        : 0
  lost        : 0             retrans        : 1
  in flight    : 2             fackets        : 0
  total retans  : 32
Times:
  last_data_sent: 2540 ms  last_data_recv: 12160 ms  last_ack_recv: 10300 ms
Keepalive:
  interval     : 7200 s         retry         : 9
Metrics:
  usable_wnd   : 65215          bw            : 9800 B/s
  snd_ssthresh  : 2             rcv_ssthresh   : 25560
  snd_cwnd     : 2             rtt           : 10 ms
  snd_wnd      : 65535          rttvar         : 50 ms
  reordering   : 3             rcv_rtt        : 0 ms
  mss_cache    : 1420          advmss         : 1420
  in_pkt_num   : 4             out_pkt_num    : 54
Queue length:
  recv_queue_len: 0        write_queue_len: 64      ofo_queue_len: 0
```

Major parameters are described as follows:

- **write_queue_len**: Specifies the length of the sending queue. Packets can be sent to a specific link only when the parameter value is not null.
- **out_pkt_num**: Specifies the number of sent packets.
- **snd_cwnd**: Specifies the sending congestion window, which defines the number of packets allowed to be sent to a link. It is an important parameter to control congestion.
- **total retans**: Specifies the number of retransmitted packets.
- **srtt**: Specifies the current delay of the link.

14) Enter line card mode to check RTP shaping statistics.

```
[LC2/1]#show rtp-shaping statistics
ref_count[1], sch_count[10671], free_count[1022]
rtp-que[4dd54cf0]:
       delay[1000],clock_rate[90000]
       flow[124.1.1.2,66.1.1.55,6,554,1776]
       que-pkts[179],pass_pkts[113],video_frames[26],audio_frames[0],disorder_frames[0]
       rtp_time[991451572],sys_time[585873838],reply_cseq[3]
```

Configuring HTTP Service

Understanding HTTP

Overview

The Hypertext Transfer Protocol (HTTP) is used to transmit web page information over the Internet. HTTP resides at the application layer of the TCP/IP protocol stack. The transmission layer uses connection-oriented TCP.

Hypertext Transfer Protocol Secure (HTTPS) is the HTTP supporting the Secure Sockets Layer (SSL). HTTPS sets up a secure channel on an insecure network to ensure that information can hardly be intercepted and to defend against man-in-the-middle attacks to some extent. Currently, HTTPS has been widely used among security-sensitive communication services, such as electronic payment.

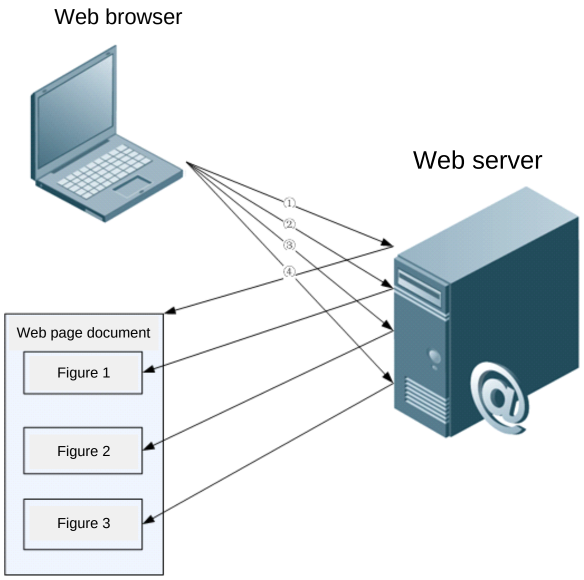
Basic Concept

HTTP Service

The HTTP service facilitates HTTP to transmit web page information over the Internet. HTTP/1.0 is the most popular HTTP version in the industry. HTTP/1.0 uses the short connection mode to simplify connection management, as a web server may be accessed for tens of thousands or even a million times each day. When receiving a connection request, the server sets up a TCP connection and releases it after the request is completed. The server does not record or trace previous requests. Although HTTP/1.0 simplifies connection management, it introduces certain performance defects.

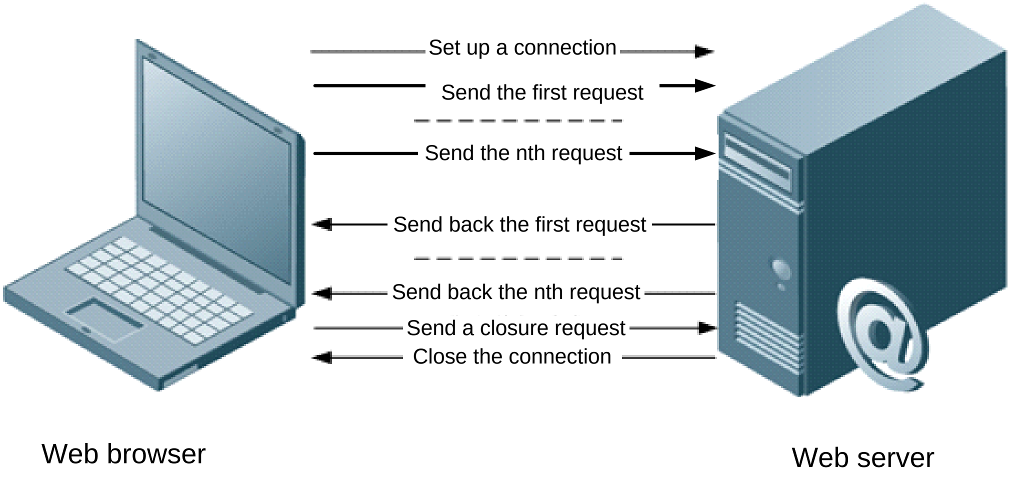
For example, a web page may contain URLs of multiple images, so that the browser sends multiple requests in the access process. When receiving a request, the server sets up an independent connection which is completely isolated from other connections. The process of setting up and releasing a connection consumes plenty of resources, and therefore has serious severe impact on the performance of the client and the server, as shown in Figure 0-1.

Figure 0-1 HTTP/1.0 Protocol Packet Exchange



HTTP/1.1, however, has solved this defect. HTTP/1.1 supports a persistent connection, through which multiple requests and responses can be transmitted. The client can send the next request before the previous request is completed, thereby reducing network delay and enhancing performance, as shown in Figure 0-2.

Figure 0-2 HTTP/1.1 Protocol Packet Exchange



Currently, Ruijiedevices support HTTP/1.0 and HTTP/1.1.

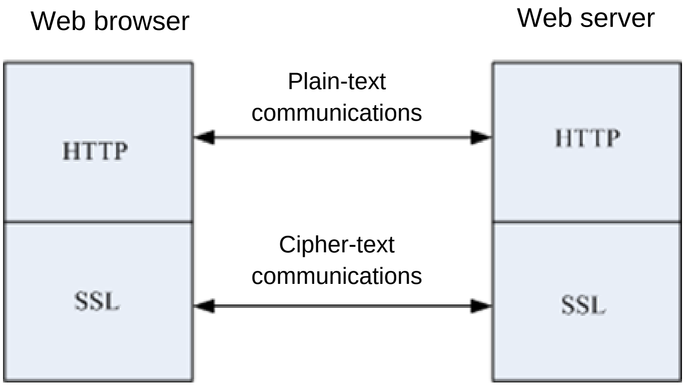
Note The protocol version used by a device depends on the specific web browser.

HTTPS Service

HTTPS adds the security base of SSL to HTTP. To enable HTTPS to run normally, the server must have a Public Key Infrastructure (PKI) certificate, which is not necessary for the client. SSL provides the following services:

- Authenticating users and servers to ensure that data is sent to correct clients and servers
- Encrypting data to prevent data interception during transmission
- Keeping data integrity to ensure that data is not changed during transmission

Figure 0-3 HTTPS Service



HTTP Upgrade Service

The HTTP upgrade service includes local and remote HTTP upgrade services.

- During local upgrade, the device works as an HTTP server. Users can log in to the device through the web browser and upload the upgrade files to the device so as to upgrade files on the device.
- During remote upgrade, the device works as a client connected to a remote HTTP server. It obtains the upgrade files from the server so as to upgrade local files.

Working Principle

HTTP Working Process

HTTP is used for web management. Users log in to the device through the web interface for configuration and management. Web management involves the web client and web server. The HTTP service adopts the client/server mode accordingly. The HTTP client is embedded in the web browser of the web management client and can send HTTP packets, receive HTTP response packets, and handle HTTP response packets. The web server (HTTP server) is embedded in the device. The client and the server exchanges information with each other according to the following process:

- The client sets up a TCP connection with the server. The default HTTP port number is 80, and the default HTTPS port number is 443.
- The client sends a request to the server.
- After processing the request, the server sends a response to the client.
- After processing a request, the HTTP service directly closes the TCP connection between the client and the server; while HTTPS can handle multiple requests until the client sends a TCP connection closure request or until the connection is closed due to server timeout.

The HTTP remote upgrade process is summarized as follows:

- The device connects to the server. In this process, the user-configured server address is preferentially used. If the connection fails, the server address in the local upgrade record file is used to establish the connection.
- The device sends the version numbers of local programs to the server.
- After resolution, the server returns a download file list.
- The device connects to file servers according to the list and downloads the upgrade files as necessary.
- The device can connect to different file servers according to the different files to be downloaded.
- The device upgrades its local files.

Protocol Specification

RFC1945 - Hypertext Transfer Protocol -- HTTP/1.0

RFC2616 - Hypertext Transfer Protocol -- HTTP/1.1

RFC2818 - Hypertext Transfer Protocol Over TLS -- HTTPS

Typical Application

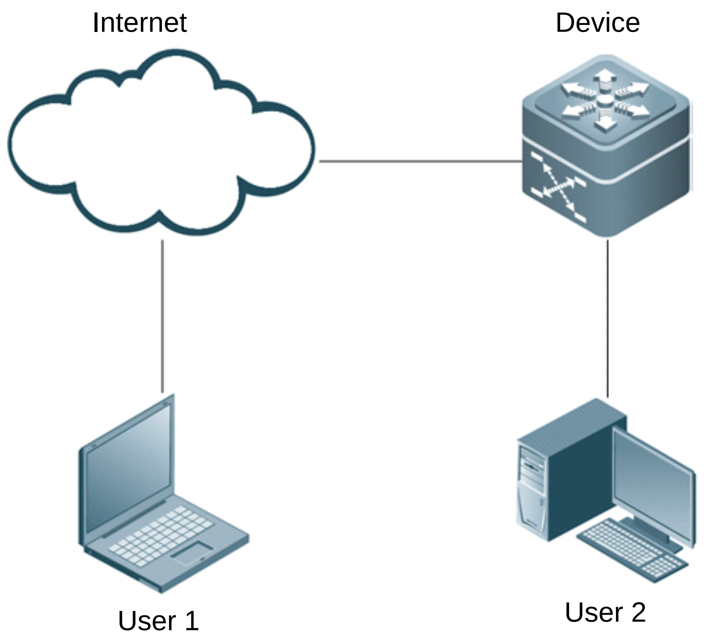
HTTP Application Service

Currently, the web NMS is still a major method for users to maintain and manage devices. Ruijie network devices also provide the web management function. When HTTP is enabled, users can log in to the web management interface after entering "http://+device IP address" on the PC browser and passing the authentication. Through the web interface, users can perform various operations, such as monitoring device states, configuring devices, uploading files, and downloading files.

The common HTTP-based service is actually insecure. For security-sensitive communications, Ruijie devices also provide the more secure HTTPS service, which encrypts the information transmitted between users and the device, so that third-party devices cannot intercept or modify the information. Users can perform web management simply after entering "https://+device IP address" on the web browser and passing the authentication.

Figure 0-4 illustrates a typical web management scenario. Users can remotely access and manage the device through the Internet or log in to the web server through a LAN to perform configuration management for the device. Users can enable either HTTPS or HTTP, or both as necessary on the device.Users can also specify HTTP/1.0 or HTTP/1.1 on the web browser for accessing the HTTP service of the device.

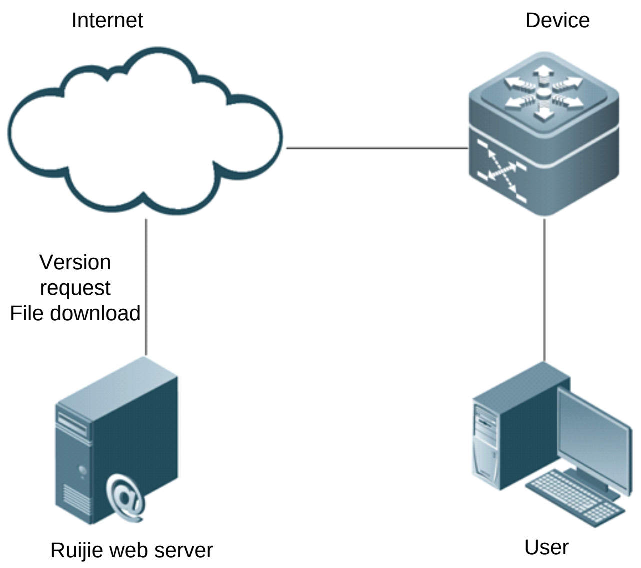
Figure 0-4 HTTP Application Scenario



HTTP Remote Upgrade Service

The HTTP Remote Upgrade Service means that a device serving as a client connects the remote HTTP server and obtains files from the server to upgrade local files. The default domain name of Ruijie web server is "rgos.ruijie.com.cn." Figure 0-5 shows a typical application scenario.

Figure 0-5 HTTP Remote Upgrade



Configuring HTTP

Default Configuration

The following table describes the default configuration of HTTP.

Feature	Default Setting
Enabling the HTTP service	The HTTP service is disabled by default.
HTTP authentication method	Usernames: **admin** and **guest**
HTTP service port	Common HTTP port number: 80 HTTPS port number: 443
HTTP upgrade server	Server address: 0.0.0.0 Port number: 80
HTTP upgrade mode	Manual
HTTP upgrade auto-detection time	Random

Prerequisites

Before configuring the domain name of the HTTP upgrade server, enable the DNS function on the device and configure the address of the DNS server.

Configuration Steps

Step	Configuration Task	Description
1	Enable the HTTP service.	Mandatory
2	Configure HTTP authentication information.	(Optional) This step is performed when authentication information needs to be modified.
3	Configure the HTTP port.	(Optional) This step is performed when the HTTP port needs to be changed.
4	Configure the HTTP upgrade server.	(Optional) This step is performed when the server address needs to be specified.

5	Configure the HTTP upgrade mode.	(Optional) This step is performed when the upgrade mode needs to be changed.
6	Configure HTTP upgrade auto-detection time.	(Optional) This step is performed when the HTTP upgrade auto-detection time needs to be changed.
7	Manually upgrade files with HTTP.	Mandatory

Enabling the HTTP Service

The HTTP service includes the commonly used HTTP service and the HTTPS service. HTTPS adds SSL on the basis of HTTP to enhance information security.

Use the following commands to enable the HTTP service in configuration mode.

Command	**Function**
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)#**enable service web-server http**	(Mandatory) Enables the HTTP service.
Ruijie(config)#**enable service web-server https**	(Mandatory) Enables the HTTPS service.
Ruijie(config)#**enable service web-server [all]**	(Mandatory) Enables both HTTP and HTTPS services.

Configuration example:

The following example enables both HTTP and HTTPS services on a Ruijie device.

```
Ruijie# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)# enable service web-server
```

Configuring HTTP Authentication Information

When HTTP is enabled, users can log in to the web interface only after being authenticated. Use the **webmaster level** command to configure HTTP authentication information.

Command	**Function**
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(confing)# **webmaster level** *privilege-level* **username** *name* **password** { *password* | [**0 | 7**] *encrypted-password* }	(Mandatory) Configures the login authentication mode, which is not configured by default.

Note

Usernames and passwords come with three permission levels, each of which includes at most 20 usernames and passwords.

Configuration example:

The following example uses the username **admin** and plain-text password **ruijie** at level 0 to perform web authentication on a Ruijie device.

```
Ruijie# configure terminal
```

```
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)# webmaster level 0 username admin password ruijie
```

Configuring the HTTP Port

Configuring the HTTP port can reduce attacks from unauthorized users to HTTP. Ruijie devices support the HTTP and HTTPS service modes.

- Use the following commands to configure the HTTP port number.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **ip http port** *port-number*	(Optional) Configures the HTTP port number, which is 80 by default.

Configuration example:

The following example configures the HTTP port number as 8080 on a Ruijie device.

```
Ruijie# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)# ip http port 8080
```

- Use the following commands to configure the HTTPS port.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **ip http secure-port** *port-number*	(Optional) Configures the HTTPS port number, which is 443 by default.

Configuration example:

The following example configures the HTTPS port number as 4430 on a Ruijie device.

```
Ruijie# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)# ip http secure-port 4430
```

Configuring the HTTP Upgrade Server

The address of the HTTP remote upgrade server is 0.0.0.0 and the port number is 80 by default. Use the following commands to change the server address.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **http update server** { *host-name* | *ip-address* } [**port** *port-number*]	(Optional) Configures the address of the HTTP upgrade server.

Note The HTTP upgrade server address does not need to be configured because the local upgrade record file records available upgrade server addresses.

Caution If the server domain needs to be configured, enable the DNS function on the device and configure the DNS server address.

Caution The server address cannot be an IPv6 address.

Configuration example:

The following example configures the domain name of the HTTP upgrade server as **rgos.ruijie.com.cn** and the port number as 85 on a Ruijie device.

```
Ruijie# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)# http update server rgos.ruijie.com.cn port 85
```

Configuring an HTTP Upgrade Mode

The manual upgrade mode applies for HTTP by default. Use the following commands to enter global configuration mode and configure HTTP to automatically detect the files available for upgrade on the server.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **http update mode auto-detect**	(Optional) Configures the HTTP upgrade mode as auto-detection. If this step is not performed or the **no** form of this command is executed, the manual upgrade mode is used by default.

In auto-detection mode, the device detects the files on the server during upgrade. Users can find the web versions to be upgraded through the web interface.

Configuration example:

The following example configures the HTTP upgrade mode as auto-detection on a Ruijie device.

```
Ruijie# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)# http update mode auto-detect
```

Configuring HTTP Upgrade Auto-Detection Time

In auto-detection mode, the remote HTTP auto-detection time is random. Use the following commands to change the auto-detection time in global configuration mode.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **http update time daily** *hh*:*mm*	(Optional) Configures HTTP auto-detection time, which is random by default.

Note

The HTTP auto-detection time is a specific time point with the accuracy of minutes each day.

Caution

This configuration command takes effect only when the HTTP upgrade mode is auto-detection.

Configuration example:

The following example configures the HTTP auto-detection time as 3:00 am on a Ruijie device.

```
Ruijie# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)# http update time daily 03:00
```

Manually Upgrading Files with HTTP

- Remote Upgrade

HTTP provides only the remote auto-detection function by default, and the system does not automatically perform upgrade. Use the following commands to upgrade the system in privileged EXEC mode.

Command	Function
Ruijie# **http check-version**	(Optional) Checks the upgrade version.
Ruijie# **http update web** [**version** *string*]	Updates the web package.

Configuration example:

The following example performs remote file upgrade for a Ruijie device through HTTP.

```
Ruijie# http check-version
app name:web
sn           version               filename
-- ------------------- -------------------------
0        1.2.1(82381)     web1.2.1(145680).upd
1        1.2.1(82380)     web1.2.1(145680).upd
2        1.2.1(82379)     web1.2.1(145680).upd
3        1.2.1(82378)     web1.2.1(145680).upd
```

- Local Upgrade

You can use the **copy tftp** command to download latest web files to a Ruijie device and then use the following command to upgrade the web package.

Command	Function

Ruijie# **http web-file update**	Updates the web package.

Caution To enable the new web package to take effect, log in to the web interface again.

The following example locally upgrades the web package for a Ruijie device.

```
Ruijie#copy tftp://10.10.10.13/web_management_pack.upd flash:web_management_pack.upd
Ruijie#http web-file update
```

Monitoring and Maintaining HTTP

Displaying HTTP Configuration Information

Command	Function
show web-server status	Displays the configuration information and status of the web service.

Configuration example:

The following example displays the HTTP configuration information of a Ruijie device.

```
Ruijie# show web-server status
http server status : enabled
http server port : 80
https server status: enabled
https server port: 443
http(s) use memory block: 768, create task num: 0
```

Configuration Examples

HTTP Configuration Example

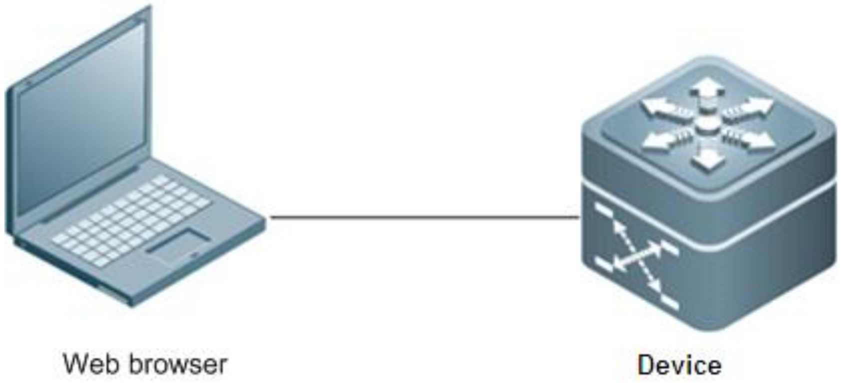
Networking Requirements

Network administrators hope to manage a device through web, and therefore log in to the device through the web browser to configure the switch.

- Log in with the user-configured authentication information.
- Ensure that the web browser can be accessed through HTTP or HTTPS so as to enhance security.
- Configure the HTTP port to reduce attacks from unauthorized users to HTTP.

Networking Topology

Figure 0-6 HTTP Application Topology



Configuration Tips

To meet the customer's requirements, focus on the following points:

- Use the **webmaster level** command to configure authentication information.
- Enable HTTP and HTTPS at the same time to meet the customer's security requirements.
- Configure the HTTP port number as 8080 and the HTTPS port number as 4430.

Configuration Steps

15) Configure the username as admin and the password as ruijie.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)# webmaster level 0 username admin password ruijie
```

16) Enable the HTTP and HTTPS services.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#enable service web-server
```

17) Configure the HTTP port number as 8080.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#ip http port 8080
```

18) Configure the HTTPS port number as 4430.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#ip http secure-port 4430
```

Verification

19) Check HTTP configuration information.

```
Ruijie#show web-server status
http server status : enabled
http server port : 8080
https server status: enabled
https server port: 4430
http(s) use memory block: 768, create task num: 0
```

Configuration Example of HTTP Remote Upgrade

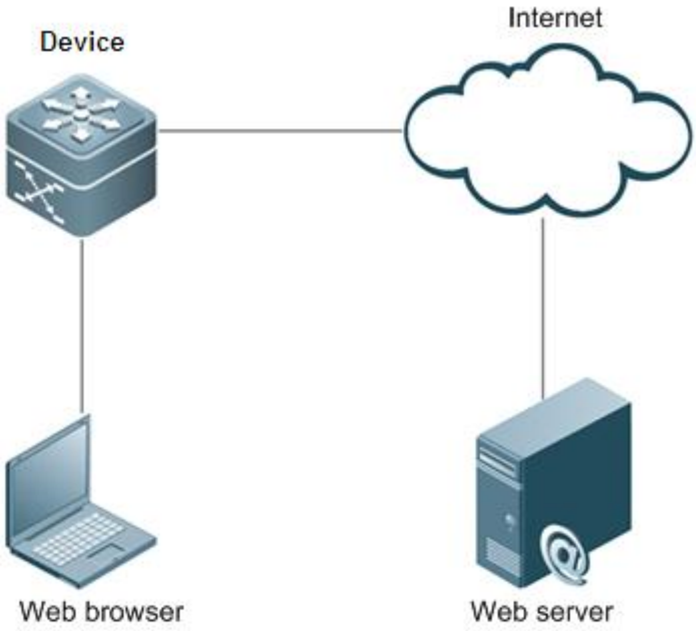
Networking Requirements

An enterprise purchasing a Ruijie device hopes to use the HTTP upgrade function to upgrade files.

- Ensure that the device can periodically and remotely obtain information about the files available for upgrade from a Ruijie server.
- Check the files currently available for upgrade.
- Download the latest files from the Ruijie server and update the device to be upgraded.

Networking Topology

Figure 0-7 Networking Topology of HTTP Remote Upgrade



Configuration Tips

To meet the customer's requirements, focus on the following point:

- Configure the device to remotely obtain information about the latest files at 2:00 am each day.

Configuration Steps

20) Configure DNS information.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
```

```
Ruijie(config)#ip domain-lookup                            //Enable the DNS function on the
device.
Ruijie(config)#ip name-server 192.168.5.134                   //Configure the IP address of
the DNS server.
```

21) Configure the address of the upgrade server.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)# http update server rgos.ruijie.com.cn
```

22) Enable the auto-detection mode and configure the remote detection time of the device as 2:00 am.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#http update mode auto-detect
Ruijie(config)#http update time daily 02:00
```

23) Obtain information about the files available for upgrade from the remote server.

```
Ruijie#http check-version
app name:web
sn           version               filename
-- ------------------- -------------------------
0       1.2.1(82381)     web1.2.1(145680).upd
1       1.2.1(82380)     web1.2.1(145680).upd
2       1.2.1(82379)     web1.2.1(145680).upd
3       1.2.1(82378)     web1.2.1(145680).upd
```

24) Download the files from the server and update the device.

```
Ruijie#http update web
```

Verification

Check server version information on the online upgrade interface of web.

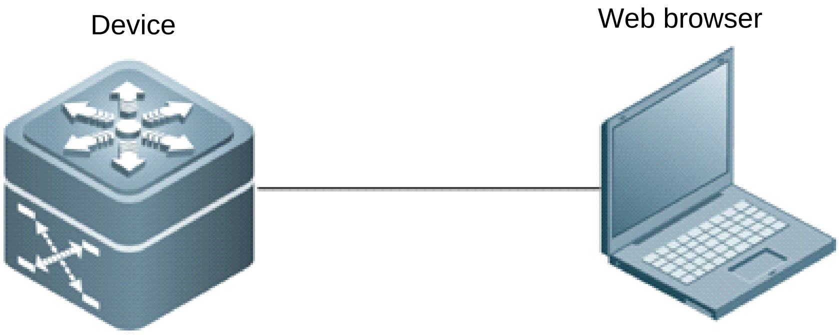
Configuration Example of HTTP Local Upgrade

Networking Requirements

- Users hope to run the latest web package, which is obtained from an official website, on a device.

Networking Topology

Figure 0-8 Networking Topology of HTTP Local Upgrade



Configuration Tips

To meet the customer's requirements, focus on the following points:

- Connect the device to a local PC whose IP address is 10.10.10.13, and configure the device with an IP address 10.10.10.131 in the same network segment.
- Download the latest web package to the device.
- Update the web package on the device.

Configuration Steps

25) Create VLAN1 and configure an IP address for the device

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#vlan 1
Ruijie(config-vlan)#exit
Ruijie(config)#interface vlan 1
Ruijie(config-VLAN 1)#ip address 10.10.10.131 255.255.255.0
```

26) Enable the TFTP server function on the PC and run the copy tftp command on the device to download the web package.

```
Ruijie#copy tftp://10.10.10.13/web_management_pack.upd flash:web_management_pack.upd
```

27) Update the web package on the device.

```
Ruijie#http web-file update
```

Verification

On the PC, log in with web authentication once again to check whether the latest web interface is displayed.

Configuring RADIUS Dynamic Authorization Extension

Understanding RADIUS Dynamic Authorization Extension

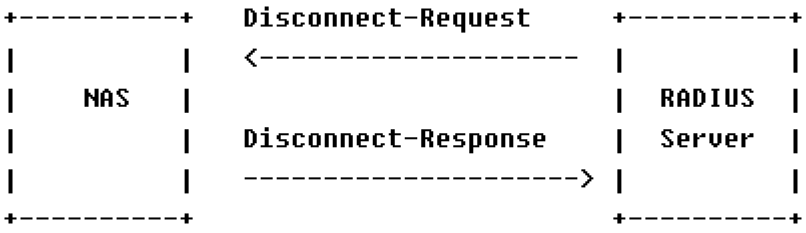
Overview

The Dynamic Authorization Extensions to Remote Authentication Dial In User Service (RADIUS) protocol is defined in RFC 3576 by IETF. This protocol defines a user offline management method, that is, the device communicates with a RADIUS server through Disconnect-Messages (DMs) to log out authenticated users. This protocol enables devices from different vendors to communicate with a RADIUS server and log out users of these devices.

The DM mechanism is as follows: A RADIUS sends user logout requests to the device. The device logs out the users that match the session IDs in the request packets, and sends response packets that contain processing results to the server. This mechanism allows the RADIUS server to manage user logout.

Working Principle

Figure 1-1 DM exchange for RADIUS dynamic authorization extension



```
+----------+   Disconnect-Request     +----------+
|          |   <--------------------  |          |
|    NAS   |                          |  RADIUS  |
|          |   Disconnect-Response    |  Server  |
|          |   ---------------------> |          |
+----------+                          +----------+
```

The above figure shows the DM exchange between the RADIUS server and device. When the RADIUS server sends a Disconnect-Request packet to the UDP port numbered 3799, the device processes the packet and sends a Disconnect-Response packet containing the processing results to the server.

Protocol Specification

RADIUS is defined in RFC 3576.

Default Configuration

The default configuration about RADIUS dynamic authorization extension is shown in the table below.

Feature	Default Setting
RADIUS dynamic authorization extension	Disabled
The number of a UDP port for intercepting DMs	3799

Configuring RADIUS Dynamic Authorization Extension

Enabling RADIUS Dynamic Authorization Extension

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)#**radius dynamic-authorization-extension enable**	Enables RADIUS dynamic authorization extension.
Ruijie(config)# **show running-config**	Shows configuration.

Use the **no radius dynamic-authorization-extension enable** command to disable RADIUS dynamic authorization extension in global configuration mode.

The example below shows how to configure RADIUS dynamic authorization extension:

```
Ruijie# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# radius dynamic-authorization-extension enable
Ruijie(config)# show run
```

Note

By default, RADIUS dynamic authorization extension is disabled.

Viewing the Configuration

Command	Function
Ruijie# **show radius dynamic-authorization-extension statistics**	Shows statistics about RADIUS dynamic authorization extension.
Ruijie# **clear radius dynamic-authorization-extension statistics**	Clears statistics about RADIUS dynamic authorization extension.

The example below shows how to show statistics about RADIUS dynamic authorization extension.

```
Ruijie# show radius dynamic-authorization-extension statistics
Disconnect-Request Received:                        50
Incorrect Disconnect-Request Received:               1
Disconnect-Request Dropped for Queue Full:            0
Disconnect-Request Process Timeout:                  0
Disconnect-Request Process Success:                  49
Disconnect-ACK Sent:                                 25
Disconnect-ACK Sent Failed:                            0
Disconnect-NAK Sent:                                 24
```

```
Disconnect-NAK Sent Failed:                          0
```

Configuring Optional Features of RADIUS Dynamic Authorization Extension

Configuring a UDP Port

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)#**radius dynamic-authorization-extension port** *num*	Sets a UDP port for intercepting the packets about RADIUS dynamic authorization extension. The value ranges from 1024 to 65535. The default value is 3799.
Ruijie(config)# **show running-config**	Shows configuration.

Use the **no radius dynamic-authorization-extension port** command to restore the default interception port in global configuration mode.

The example below shows how to configure a UDP port intercepting the packets about RADIUS dynamic authorization extension.

# Set the port numbered 8080 to intercept RADIUS requests:

```
Ruijie# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# radius dynamic-authorization-extension port 8080
Ruijie(config)# show running-config
```

# Reset the configuration:

```
Ruijie# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# no  radius dynamic-authorization-extension port
Ruijie(config)# show running-config
```

Examples for Configuring RADIUS Dynamic Authorization Extension

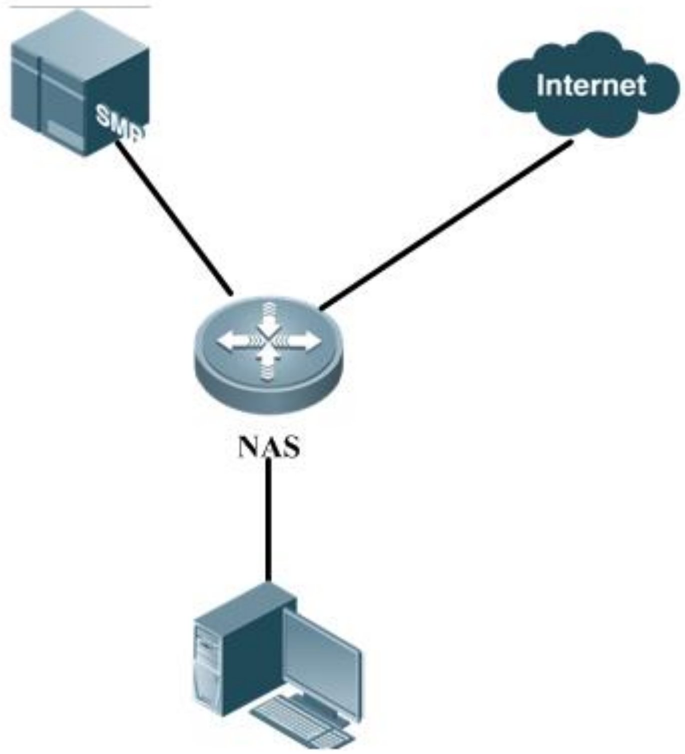
Networking Requirements

RADIUS dynamic authorization extension must work with the authentication mechanism. The network comprises SAM servers, Ruijie access devices, and PCs of users.

Ruijie access devices must support RADIUS dynamic authorization extension.

Networking Topology

Figure 4-1 Network topology of RADIUS dynamic authorization extension



Configuration Procedure

1. Configure AAA on the access authentication device.

```
aaa new-model
aaa accounting update periodic 1
aaa accounting update
aaa accounting network default start-stop group radius
aaa authentication ppp default local group radius

```

2. Configure the PPP authentication on the access device.

```
vpdn enable
vpdn-group 1
accept-dialin
protocol l2tp
  virtual-template 1

interface Virtual-Template 1
ppp authentication chap
 ip unnumbered Loopback 1
interface Loopback 1
 ip address 110.1.1.254 255.255.255.0
interface GigabitEthernet 8/1/1
```

```
ip address 100.1.1.2 255.255.255.0
duplex auto
     speed auto
```

3. Enable RADIUS dynamic authorization extension.

```
radius dynamic-authorization-extension enable
```

4. After the user logs in, the administrator uses RADIUS dynamic authorization extension on the SAM to log out the user.
5. The user is logged out and needs to be authenticated again to access the network.

Configuring Smart Status Monitoring

Overview

Introduction

A network egress generally provides two or more operator links. For example, in a campus network of a university, a network egress provides an education network link and an operator link (China Telecom, China Unicom, and China Mobile); in a government network, a network egress provides links of China Telecom, China Unicom, and China Mobile. When links in a wide area network (WAN) have different bandwidths and the number of links is no less than three, there can be 2^n adjustment policies, and, it is impossible to manually perform adjustment at any time. In view of this, the smart status monitoring module can sense network changes, and can perform complex logical operation, to obtain the final policy and implement the corresponding control script.

For example, in an aggregate port (AP) scenario where all AP members operate properly, the HQoS (Hierarchical Quality of Service) traffic policy needs to limit the traffic rate based on the sum of bandwidth of all AP members and guarantee user services based on priorities. When the link of a certain AP member is down, the HQoS traffic policy changes accordingly. In this case, the total bandwidth to be guaranteed is the sum of bandwidth of all AP members subtracted by the bandwidth of the failed member link.

Working Principle

The smart status monitoring module loads and runs different scripts to control the device according to different preset status detection situations of scheduling instances.

Detection events mainly involve the BDF session status, interface link status, the selected, unselected, and track the status of Link Aggregation Control Protocol (LACP) on the AP members. According to the configured smart status monitoring instances, unary operation or binary operation between multiple detection events are performed starting from the predetermined operation sequence. If the operation result meets the detection quantity, the smart status monitoring module loads and runs the script copied by TFTP to control the device.

Different scripts are executed according to different detection results. In this way, the device can be flexibly controlled, thereby reducing the maintenance workload.

Defaults

Feature	Default Value
Smart Status Monitoring Scheduling Instance	N/A
Smart Status Monitoring Operation Sequence	N/A
Smart Status Monitoring Detection Events	N/A
Smart Status Monitoring Heartbeat Period	5 seconds

Smart Status Monitoring Heartbeat Period

Configuring Heartbeat Period

Command	Function
config	Enters the global configuration mode.
smart-monitor tick *seconds*	Sets the scheduling heartbeat period for smart status monitoring in seconds. The value range is from **1** to **60**.
exit	Exits the global configuration mode and enters the privileged EXEC mode.

Run the **no smart-monitor tick** command to restore the default scheduling heartbeat period.

Configuration Example

#Configure the smart status monitoring heartbeat period to 2 seconds.

```
Ruijie#configure
Ruijie(config)# smart-monitor status tick 2
Ruijie(config)#exit
Ruijie#
```

Smart Status Monitoring Detection Events

Configuring Detection Event

Command	Function
config	Enters the global configuration mode.
smart-monitor status *status-id* { bfd ***peer-address*** {down **|** up**}** } [member-interface ***interface-name***]} | { **interface** *interface-name* **{ line {down | up}** | **lacp {selected | unselected}**| **link-quality { enable | disable}** *link-quality -value* } } | { **track** *track-id* {**up | down**} | { **time-range** *time-range-name* {**active | inactive** } }	Configures smart status monitoring detection events. The event ID range is from **1** to **64**. bfd *peer-address*: Indicates the peer address of the detected BFD session. down: Indicates that the detected BFD session is in the down state. up: Indicates that the detected BFD session is in the up state. *interface-name*: Indicates the name of an AP member that supports BFD session detection. *interface-name*: Indicates the name of the detected interface. down: Indicates that the line of the detected interface is down.

	up: Indicates that the line of the detected interface is up. selected: LACP of the detected interface is selected. unselected: LACP of the detected interface is not selected. *link-quality–value*: Indicates the link quality value of the detected interface. The value range is from **0** to **100**. *track-id*: Indicates detected track ID. The value range is from **1** to **700**. *time-range-name*: Indicates the name of the detected time range. active: Indicates that the detected time range is in the active state. inactive: Indicates that the detected time range is in the inactive state.
exit	Exits the global configuration mode and enters the privileged EXEC mode.

Run the **no smart-monitor status** *status-id* command to delete a smart status monitoring detection event.

Configuration Example

#Create Smart Status Detection Event 1 to detect the event that the BFD session with the peer IP address being 19.19.19.1 is up.

```
Ruijie#configure
Ruijie(config)# smart-monitor status 1 bfd 19.19.19.1 up
Ruijie(config)#exit
Ruijie#
```

#Create Smart Status Detection Event 3 to detect the event that the link of GigabitEthernet 0/0 is up.

```
Ruijie#configure
Ruijie(config)# smart-monitor status 3 interface gigabitEthernet0/0 line up
Ruijie(config)#exit
Ruijie#
```

#Create Smart Status Detection Event 4 to detect the event with the track ID being 1 and the state being up.

```
Ruijie#configure
Ruijie(config)# smart-monitor status 4 track 1 up
Ruijie(config)#exit
Ruijie#
```

#Create Smart Status Detection Event 5 to detect the event that the BFD session of the AP member Gi0/0 with the peer address being 19.19.19.1 is up.

```
Ruijie#configure
```

```
Ruijie(config)#interface Route-aggregateport 1
Ruijie(config-if-Route-aggregateport 1)# ip address 19.19.19.1 255.255.255.252
Ruijie(config-if-Route-aggregateport 1)# bfd interval 50 min_rx 50 multiplier 3
Ruijie(config-if-Route-aggregateport 1)# member interface GigabitEthernet 0/0 bind bfd
peer-ip 192.168.1.10
Ruijie(config-if-Route-aggregateport 1)# member interface GigabitEthernet 0/1 bind bfd
peer-ip 192.168.1.10
Ruijie(config-if-Route-aggregateport 1)#end
Ruijie#configure
Ruijie(config)# smart-monitor status 5 bfd 19.19.19.1 up member-interface
GigabitEthernet 0/0
Ruijie(config)#
```

#Create Smart Status Detection Event 6 to detect the event that the weekdays time range is active.

```
Ruijie(config)# smart-monitor status 6 time-range check active
```

#Create Smart Status Detection Event 7 to detect the event that the weekdays time range is inactive.

```
Ruijie(config)# smart-monitor status 7 time-range check inactive
```

#Create a time range named weekdays.

```
Ruijie(config)# time-range check
Ruijie(config-time-range)# periodic weekdays 8:30 to 17:30
```

To configure a time range, perform the following commands in privileged EXEC mode:

Command	Function
configure terminal	Enters the global configuration mode.
time-range *time-range-name*	Names a time range using a significant string.
Ruijie(config-time-range)# **absolute** [start time date] end time date	(Optional) Configures an absolute time range. For more information, see time range configuration manual.
Ruijie(config-time-range)# **periodic day-of-the-week** time to [day-of-the-week] time	(Optional) Configures a periodical time range. For details, see the time range configuration guide.
exit	Exits the global configuration mode and enters the privileged EXEC mode.

Displaying

Run the following command to display information about smart status monitoring detection events.

Command	**Function**
show smart-monitor status [status-id]	Displays information about smart status monitoring detection events. The result of a smart status monitoring detection event may be **True**, **False**, or **Fail**. **True** indicates a match, **False** indicates a mismatch, and **Fail** indicates that the configuration is incorrect and does not meet the detection target.

The displayed information is as follow:

```
Ruijie(config)# show smart-monitor status 3
Smart-monitor status 3 : final-status True
smart-monitor status 3 interface GigabitEthernet 0/0 line up
Ruijie(config)# show smart-monitor status 5
Smart-monitor status 5 : final-status False
smart-monitor status 5 bfd 19.19.19.1 up member-interface GigabitEthernet 0/0
```

Smart Status Monitoring Operation Sequence

Configuring Detection Event

Command	**Function**
config	Enters the global configuration mode.
smart-monitor status *status-id* { bfd ***peer-address*** {down **|** up} } | { **interface** *interface-name* { **line {down | up}** | **lacp {selected** | **unselected} | link-quality { enable** | **disable}** *link-quality -value* } } | { **track** *track-id* {**up | down**} **|** { **time-range** *time-range-name* {**active** | in**active** } }	Configures smart status monitoring detection events.
exit	Exits the global configuration mode and enters the privileged EXEC mode.

Run the **no smart-monitor** status ***status-id*** command to delete smart status monitoring detection events.

Configuring Operation Sequence

Command	**Function**
config	Enters the global configuration mode.

smart-monitor seq *seq-id* {**seq [not]** *seq-id*} **|** {**status [not]** *status-id*} **[** {**and | or**} {**seq [not]** *seq-id*} **|** {**status [not]** *status-id*} **]**	Configures the operation sequence for smart status detection. The operation sequence ID range is from **1** to **100**. (Optional) not: Performs the NOT operation via operators. *status-id*: Indicates the ID of a smart status monitoring detection event. The value range is from **1** to **64**. and: Indicates binary operation. The AND operation is performed via the left and right operators. or: Indicates binary operation. The OR operation is performed via the left and right operators.
exit	Exits the global configuration mode and enters the privileged EXEC mode.
show smart-monitor seq [*seq-id*]	Displays the smart status monitoring operation sequence.

Run the **no smart-monitor seq** *seq-id* command to delete a smart status monitoring operation sequence.

Configuration Example

#Configure the operation rule of Operation Sequence 1: perform negation operation on the result of Event 1 that the link of the detected interface GigabitEthernet1/1/0 is up.

```
Ruijie(config)# smart-monitor status 1 interface GigabitEthernet 0/0 line up
Ruijie(config)# smart-monitor seq 1 status not 1
```

#Configure the operation rule of Operation Sequence 2: perform binary operation, that is, perform AND operation on the statuses of Instance 2 and Instance 1 of smart status monitoring detection events.

```
Ruijie#configure
Ruijie(config)# smart-monitor status 2 interface GigabitEthernet 0/1 line up
Ruijie(config)# smart-monitor seq 2 status  2 and seq  1
```

Displaying

Run the following command to display the smart status monitoring operation sequence.

Command	Function
show smart-monitor seq *[seq-id]]*	Displays the smart status monitoring operation sequence.

The displayed information is as follow:

Run the **show smart-monitor seq** command to display all operation sequence information.

```
Ruijie#show smart-monitor schedule
```

```
smart-monitor seq 1 status not 1
smart-monitor seq 2 status  1 and seq  1
```

Smart Status Monitoring Scheduling Instance

Configuring Detection Event

Command	**Function**
config	Enters the global configuration mode.
smart-monitor status *status-id* { bfd ***peer-address*** {down **|** up**}** } } | { **interface** *interface-name* { **line** {**down | up**} | **lacp {selected** | **unselected}** | **link-quality { enable | disable}** *link-quality -value* } } | { **track** *track-id* {**up | down**} | { **time-range** *time-range-name* {**active** | **inactive** } }	Configures smart status monitoring detection events. The event ID range is from **1** to **64**. bfd *peer-address*: Indicates the peer address of the detected BFD session. down: Indicates that the detected BFD session is in the down state. up: Indicates that the detected BFD session is in the up state. *interface-name*: Indicates the name of the detected interface. down: Indicates that the line of the detected interface is down. up: Indicates that the line of the detected interface is up. selected: LACP of the detected interface is selected. unselected: LACP of the detected interface is not selected. *link-quality–value*: Indicates the link quality value of the detected interface. The value range is from **0** to **100**. *track-id*: Indicates detected track ID. The value range is from **1** to **700**. *time-range-name*: Indicates the name of the detected time range. active: Indicates that the detected time range is in the active state. inactive: indicates that the detected time range is in the inactive state.
exit	Exits the global configuration mode and enters the privileged EXEC mode.

Run the **no smart-monitor status** *status-id* command to delete a smart status monitoring detection event.

Configuring Operation Sequence

Command	Function
config	Enters the global configuration mode.
smart-monitor seq *seq-id* {**seq [not]** *seq-id*} **|** {**status [not]** *status-id*} **[** {**and | or**} {**seq [not]** *seq-id*} **|** {**status [not]** *status-id*} **]**	Configures the operation sequence for smart status detection. The operation sequence ID range is from **1** to **100**. (Optional) not: Performs the NOT operation via operators. *status-id*: Indicates the ID of a smart status monitoring detection event. The value range is from **1** to **64**. and: Indicates binary operation. The AND operation is performed via the left and right operators. or: Indicates binary operation. The OR operation is performed via the left and right operators.
exit	Exits the global configuration mode and enters the privileged EXEC mode.
show smart-monitor seq [*seq-id*]	Displays the smart status monitoring operation sequence.

Run the **no smart-monitor seq** *seq-id* command to delete a smart status monitoring operation sequence.

Configuring Scheduling Instance

Command	Function
config	Enters the global configuration mode.

smart-monitor schedule *schedule-id* **final-seq** *seq-id* **action load** *file-name tag_infor* [**times** *times-number*]	Configures a smart status monitoring scheduling instance. *schedule-id*: Indicates the serial number of a smart status monitoring scheduling instance. The value range is from **1** to **8**. *seq-id*: Indicates the serial number of an operator. The value range is from **1** to **100**.*file-name*: Indicates the name of a to-be-loaded script file. The file name contains 1 to 63 characters. *tag_infor*: Executes tagged information in the script. (Optional) times-number: Loads and runs the script labeled by **file-name** when the number of detection times reaches the value of **times-number**. The value range is from **1** to **20**. The default value is **1**.
exit	Exits the global configuration mode and enters the privileged EXEC mode.
show smart-monitor schedule [*schedule-id* [**do**]]	Displays information about smart status monitoring scheduling instances.

Run the **no smart-monitor schedule** *schedule-id* command to delete a smart status monitoring scheduling instance.

Configuration Example

#Create a smart status monitoring scheduling instance with the ID 1, perform operation starting from Operation Sequence 2, and load content of the switch_to_pos section in the configuration script ah-wh-pos-mstp-chg.text if the number of detection times reaches 3.

```
Ruijie#configure
Ruijie (config)# smart-monitor schedule 1 final-seq 2 action load
ah-wh-pos-mstp-chg.text switch_to_pos times 3
```

Syntax of the configuration file ah-wh-pos-mstp-chg.text:

```
@@@@@start
switch_to_pos:
enable
config
interface GigabitEthernet 1/4/2/7
shutdown
end
@@@@@end
```

1. Each execution section starts with @@@@@start and ends with @@@@@end.Content to be loaded is identified the section identifier in the **smart-monitor schedule 1** command.

It should be noted that the colon behind the section identifier is an English character.

Displaying

Run the following command to display information about smart status monitoring scheduling instances.

Command	Function
show smart-monitor schedule [schedule-id [do]]	Displays the smart status monitoring configuration information, automatic loading and execution information, and operation result.

The displayed information is as follow:

Example 1: Run the **show smart-monitor schedule** command to display information about all operation scheduling instances.

```
Ruijie#show smart-monitor schedule
smart-monitor schedule 1 final-seq 2 action load ah-wh-pos-mstp-chg.text switch_to_pos
times 3
smart-monitor schedule 2 final-seq 1 action load ah-wh-pos-mstp-chg.text pos_to_switch
times 1
smart-monitor schedule 3 final-seq 1 action load 3.text times 2
```

Example 2: Run the **show smart-monitor schedule** 2 command to display information about a specified operation scheduling instance.

```
Ruijie#show smart-monitor schedule 2
smart-monitor schedule 2 final-seq 1 action load ah-wh-pos-mstp-chg.text pos_to_switch
times 1
```

Example 3: Run the **show smart-monitor schedule** 2 **do** command to display information about a specified operation scheduling instance.

```
Ruijie#show smart-monitor schedule 2 do
smart-monitor schedule 2 final-seq 1 action load ah-wh-pos-mstp-chg.text pos_to_switch
times 1
```

Smart Status Monitoring Configuration Example

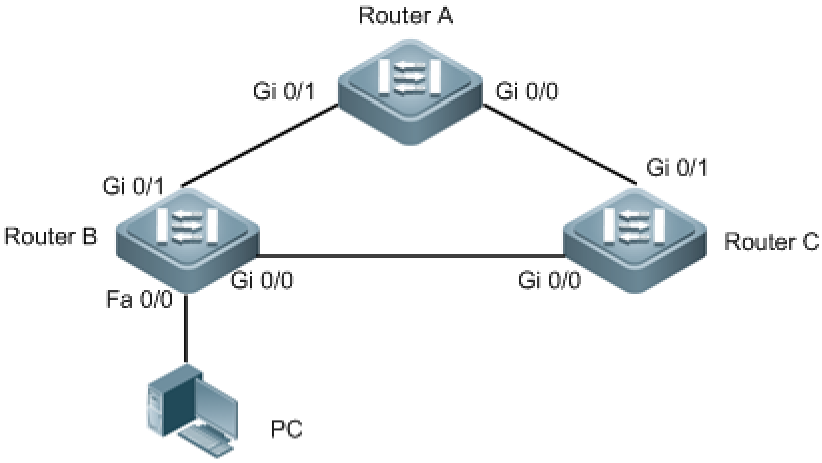
Configuration Example 1

Networking Requirements

Three routers are deployed.

Network Topology

Figure 0-1 Smart Status Monitoring



Configuration Key Points

1. Configure the flow queue, user queue, and user group queue.

2. Configure the traffic classifier, traffic behavior, and traffic policy.

3. Apply the HQoS policy.

4. Configure an AP and use LACP for detection on the AP.

5. Configure the smart status monitoring detection and dynamically adjust the HQoS policy.

Configuration Steps

#Configure the flow queue of the production service.

```
flow-queue sc_0/0
 queue be wfq weight 10
 queue af1 wfq weight 20
 queue af2 pq shaping 2666
 queue af3 wfq weight 15
 queue af4 wfq weight 15
 queue ef pq
 queue cs6 pq
 queue cs7 pq

flow-queue sc_0/1
 queue be wfq weight 10
 queue af1 wfq weight 20
 queue af2 pq shaping 5333
 queue af3 wfq weight 15
 queue af4 wfq weight 15
 queue ef pq
 queue cs6 pq
 queue cs7 pq
```

#Configure the flow queue of the office service.

```
flow-queue bg_0/1
 queue be wfq weight 10
 queue af1 wfq weight 20
 queue af2 pq shaping 2666
 queue af3 wfq weight 15
 queue af4 wfq weight 15
 queue ef pq
 queue cs6 pq
 queue cs7 pq

flow-queue bg_0/0
 queue be wfq weight 10
 queue af1 wfq weight 20
 queue af2 pq shaping 1333
 queue af3 wfq weight 15
 queue af4 wfq weight 15
 queue ef pq
 queue cs6 pq
 queue cs7 pq
```

#Configure the user queue of the production service.

```
user-queue sc_0/0 outbound
 cir 6666 pir 10000
 flow-queue sc_0/0
 user-group-queue sc_bg_0/0

user-queue sc_0/1 outbound
 cir 13333 pir 20000
 flow-queue sc_0/1
 user-group-queue sc_bg_0/1
```

#Configure the user queue of the office service.

```
user-queue bg_0/0 outbound
 cir 3333 pir 10000
 flow-queue bg_0/0
 user-group-queue sc_bg_0/0

user-queue bg_0/1 outbound
 cir 6666 pir 20000
 flow-queue bg_0/1
 user-group-queue sc_bg_0/1
```

#Configure the user group queue.

```
user-group-queue sc_bg_0/0 outbound
 shaping 10000
user-group-queue sc_bg_0/1 outbound
 shaping 20000
```

#Configure the traffic classifier of the production service.

```
traffic classifier sc_youwu_cos or
 if-match acl sc_youwu
traffic classifier sc_jinrong_cos or
 if-match acl sc_jinrong
traffic classifier sc_sudi_cos or
 if-match acl sc_sudi

traffic classifier sc_0/0 and
 if-match dst-if GigabitEthernet 0/0
 if-match src-if VLAN 2
traffic classifier sc_0/1 and
 if-match dst-if GigabitEthernet 0/1
 if-match src-if VLAN 2
```

#Configure the traffic classifier of the office service.

```
traffic classifier bg_bg_cos or
 if-match acl bg_bg
traffic classifier bg_voip_cos or
 if-match acl bg_voip
traffic classifier bg_qita_cos or
 if-match acl bg_qita

traffic classifier bg_0/0 and
 if-match dst-if GigabitEthernet 0/0
 if-match src-if VLAN 3
traffic classifier bg_0/1 and
 if-match dst-if GigabitEthernet 0/1
 if-match src-if VLAN 3
```

#Configure traffic behaviors of three sub services of the production service.

```
traffic behavior sc_jinrong_cos
 service-class be color green
traffic behavior sc_sudi_cos
 service-class af1 color green
traffic behavior sc_youwu_cos
 service-class af2 color green
```

#Configure traffic behaviors of three sub services of the office services.

```
traffic behavior bg_qita_cos
 service-class be color green
traffic behavior bg_voip_cos
 service-class af1 color green
traffic behavior bg_bg_cos
 service-class af2 color green
```

#Configure traffic behaviors of the production service.

```
traffic behavior sc_0/0
 user-queue sc_0/0 outbound
traffic behavior sc_0/1
 user-queue sc_0/1 outbound
```

#Configure traffic behaviors of the office service.

```
traffic behavior bg_0/0
 user-queue bg_0/0 outbound
traffic behavior bg_0/1
 user-queue bg_0/1 outbound
```

#Configure traffic policies of all services.

```
traffic policy sc_bg_out
 classifier sc_0/0 behavior sc_0/0 precedence 1
 classifier sc_0/1 behavior sc_0/1 precedence 2
 classifier bg_0/0 behavior bg_0/0 precedence 3
 classifier bg_0/1 behavior bg_0/1 precedence 4
```

#Configure traffic policies of three sub services of the production service.

```
traffic policy sheng_chan_in
 classifier sc_youwu_cos behavior sc_youwu_cos precedence 1
 classifier sc_jinrong_cos behavior sc_jinrong_cos precedence 2
 classifier sc_sudi_cos behavior sc_sudi_cos precedence 3
```

#Configure traffic policies of three sub services of the office service.

```
traffic policy ban_gong_in
 classifier bg_bg_cos behavior bg_bg_cos precedence 1
 classifier bg_voip_cos behavior bg_voip_cos precedence 2
 classifier bg_qita_cos behavior bg_qita_cos precedence 3
```

#Apply the HQoS policy to the AP.

```
interface AggregatePort 1
 port-user-key dst-ip
 traffic-policy sc_bg_out outbound
```

#Apply the HQoS policy to AP members and configure LACP detection.

```
interface GigabitEthernet 0/0
 port-bandwidth 10000
 port-group 1 mode active

interface GigabitEthernet 0/1
 port-bandwidth 20000
 port-group 1 mode active
```

#Configure the smart status monitoring function to detect the event that LACP of the AP member Gi0/0 is selected or not selected.

```
smart-monitor status 1 interface GigabitEthernet 0/0 lacp unselected
smart-monitor seq 1 status  1
smart-monitor seq 2 status not 1
smart-monitor schedule 1 final-seq 1 action load hqos.txt  up2down times 10
smart-monitor schedule 2 final-seq 2 action load hqos.txt down2up times 10
smart-monitor tick 10
```

#Compile the corresponding script for configuring the smart status monitoring function to detect the event that LACP of the AP member Gi0/0 is selected or not selected.

//Smart status monitoring script

```
@@@@@start
up2down:
enable
config
interface GigabitEthernet 0/0
shutdown
end
@@@@@end
@@@@@start
Down2up:
enable
```

```
config
interface GigabitEthernet 0/0
no shutdown
end
```

Verification

1. Display the smart status monitoring information on the device.

```
Ruijie# show smart-monitor status 1
Smart-monitor status 1 : final-status True
smart-monitor status 1 interface GigabitEthernet 0/0 line up

Ruijie#show smart-monitor schedule 1
smart-monitor schedule 1 final-seq 1 action load hqos.txt up2down times 3
```

Configuring Domain Name Policy

Overview

Introduction

On current routers, policy functions such as routing and flow control are implemented based on IP addresses. Although such processing is quick and accurate, an IP address involved in the processing is abstract to users and it is inconvenient to record IP addresses.

Policy routing provides a new bridge between service modules and users. Original configurations and logics of the service modules are retained and users only need to record domain names (URLs) that are easy to memorize.

Basic Concepts

URL Library

The URL library is generated in the device memory based on the URL library file and the URLs configured by the user in URL classes. The URL library includes specific domain names and domain name classification information.

URL Class

A URL class is a group of URLs with the same attribute.

URL Group

A URL group is a policy that combines a URL class and a user group.

DNS Parser

The DNS parser parses DNS response packets in depth, to obtain the domain names and corresponding IP addresses from the packets.

Working Principle

A URL group is configured on the router to specify the association between a URL class and a user group. After receiving a DNS response packet, the router searches the URL library to obtain the URL class and URL group that are corresponding to the URL in the DNS response packet, so as to further obtain the corresponding user group. The router can fill the answer IP address in the DNS response packet into the user group, and apply the user group to policy-based routing (PBR), thereby implementing routing based on the domain name.

The ultimate effect of the domain name policy function is to fill IP addresses corresponding to a type of domain names into a user group. For details about how to user this user group, refer to related configurations of other functional modules.

Application

The domain name policy module is a public module that must operate in combination with other modules to meet requirements of corresponding scenarios. For example, the domain name policy module can cooperate with HQoS to provide domain name-based service guarantee, or can associate with PBR to provide domain name-based policy routing.

Domain Name Policy Configuration

Defaults

No domain name policy is configured by default.

Configuration Prerequisites

Configuration Steps

Step	Task	Description
1	Enable the URL library.	Mandatory.
2	Configure the URL class.	Optional.
3	Configure the user group.	Mandatory.
4	Configure the URL group.	Mandatory.
5	Enable the DNS parser.	Mandatory.

Enabling URL Library

Enable the URL library by running the **lib.dat** file in the **flash:/urlib/lib.dat** path on the device. If the URL library file does not exist on the device, obtain this file and save it to the corresponding path.

Similar to other files, the library file can be downloaded to the device in TFTP or FTP mode, for example, copy the file from the **tftp://101.101.101.155/lib.dat** path or the **flash:/urlib/lib.dat** path.

Command	Function
Ruijie(config)#url-lib enable	Enables the URL library.

Configuring URL Class (Optional)

Run the following commands to configure a URL class. Because the system presets several URL classes in the URL library, this function is optional.

Command	Function
Ruijie(config)#url-class *class_name*	Enters the URL class configuration mode.
Ruijie(config-url-class)#url *url*	Adds a URL class.

Example: Configure URL redirection rules on physical interfaces.

```
Ruijie#configure terminal
Ruijie(config)# url-class test_class
Ruijie(config-url-class)#url www.ruijie.com.cn
Ruijie(config-url-class)#url *.edu.cn
```

Configuring User Group

Run the following command to configure a user group.

Command	Function
Ruijie(config)#user-group *groupname*	Configures a user group.

Example: Configure URL redirection rules on physical interfaces.

Configuring URL Group

Run the following command to configure a URL group:

Command	Function
Ruijie(config)#url-group *num*	Enters the URL group configuration mode.
Ruijie(config-url-group)#class *class_name*	Configures forward proxy mapping.
Ruijie(config-url-group)#relate user-group *group_name*	Configures a user group associated with the URL group.

Example: Configure URL redirection rules on physical interfaces.

```
Ruijie#configure terminal
Ruijie(config)# url-group 3
Ruijie(config-url-group)#class test_class
Ruijie(config-url-group)#relate user-group test_group
Ruijie(config-url-group)#exit
Ruijie(config)#
```

Enabling DNS Parser

Run the following command to enable the DNS parser on an interface:

Command	Function
Ruijie(config-if)# dns-parser enable	Enables DNS forward proxy on an interface.

Example: Configure URL redirection rules on physical interfaces.

```
Ruijie#configure terminal
Ruijie(config)#interface gigabitEthernet 1/1/1
Ruijie(config-if-GigabitEthernet 1/1/1)#dns-parser enable
Ruijie(config-if-GigabitEthernet 1/1/1)#exit
Ruijie(config)#
```

Monitoring

Run the following command to display the current URL group list.

Command	Function
Ruijie# show url-group	Displays the current URL group list.

Configuration Example

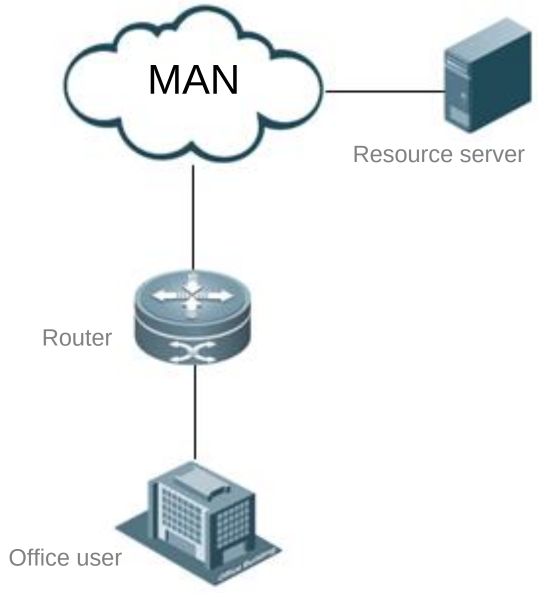
Domain Name-based Service Guarantee

Networking Requirements

A resource server is deployed in a MAN and can be accessed at www.XXX.net. The network quality of office users must be guaranteed when they access (for example, video resources in) the resource server in the MAN.

Network Topology

Figure 2-1



The office network is connected to the MAN via a router. The router is connected to the office network via GigabitEthernet0/0.

Configuration Steps

1) Configure the basic network connection.

Configure basic egresses and routes on the router based on different routing requirements and scenarios.

2) Configure the domain name policy.

```
Ruijie(config)#user-group ziyuan
Ruijie(config-user-group)#exit
Ruijie(config)#
Ruijie(config)#url-lib enable
Ruijie(config)#url-class ziyuan
Ruijie(config-url-class)#url *.XXX.net
Ruijie(config-url-class)#exit
Ruijie(config)#
Ruijie(config)# url-group 5
Ruijie(config-url-group)#class ziyuan
Ruijie(config-url-group)#relate user-group ziyuan
Ruijie(config-url-group)#exit
Ruijie(config)#
Ruijie(config)#url-group ttl 60
```

3) Configure the HQoS

The HQoS configuration is flexible. For details, see the configuration guides of router-related products and the HQoS chapter in the command reference document. In this document, HQoS is applied to the outbound direction of the inside interface of the port queue is used as an examples.

```
Ruijie(config)# ip access-list extended 100
Ruijie(config-ext-nacl)# permit ip user-group ziyuan any
Ruijie(config-ext-nacl)#exit
Ruijie(config)#
Ruijie(config)#traffic classifier aaa
Ruijie(config-traffic-classifier)#if-match acl 100
Ruijie(config-traffic-classifier)#exit
Ruijie(config)#
Ruijie(config)#traffic behavior aaa
Ruijie(config-traffic-behavior)#service-class cs6 color green
Ruijie(config-traffic-behavior)#exit
Ruijie(config)#
Ruijie(config)#traffic policy aaa
Ruijie(config-traffic-policy)#classifier aaa behavior aaa
Ruijie(config-traffic-policy)#exit
Ruijie(config)#
Ruijie(config)#port-queue test
Ruijie(config-port-queue)#exit
Ruijie(config)#
Ruijie(config)#interface gigabitEthernet 0/0
Ruijie(config-if-GigabitEthernet 0/0)#traffic-policy aaa outbound
Ruijie(config-if-GigabitEthernet 0/0)#port-queue test shaping 2000
Ruijie(config-if-GigabitEthernet 0/0)#exit
Ruijie(config)#
```

4) Enable the DNS parser and DNS proxy on the interface.

```
Ruijie(config)#interface GigabitEthernet 0/0
Ruijie(config-if-GigabitEthernet 0/0)#dns-parser enable
Ruijie(config-if-GigabitEthernet 0/0)#exit
Ruijie(config)#
```

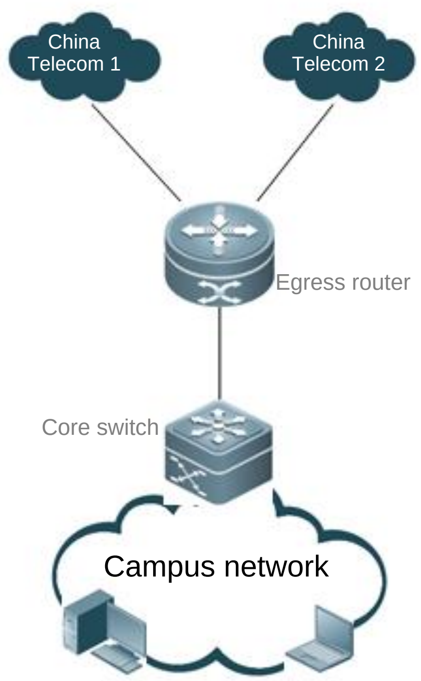
URL Routing

Networking Requirements

A campus network provides two China Telecom egresses: China Telecom 1 and China Telecom 2. All services are connected to the public network via China Telecom 2. In this example, it is required to connect the CNKI service to the public network via China Telecom 1.

Network Topology

Figure 2-2



Connection Between Interface TenGi1/1/1 and Core Switch of Campus Network

Configuration Steps

(1). Configure outbound routes for operators.

According to different routing requirements, intranet user packets are forwarded on different operator links based on operator routes, PBR routes, user routes, and the like. DNS packets as user packets are also routed based on the same policy.

(2). Configure the domain name policy.

```
Ruijie(config)#url-lib enable
Ruijie(config)#user-group cnki
Ruijie(config-user-group)#exit
Ruijie(config)#
Ruijie(config-url-class)#url *.cnki.net
Ruijie(config-url-class)#url cnki.*
Ruijie(config-url-class)#uexit
Ruijie(config)#
Ruijie(config)# url-group 3
Ruijie(config-url-group)#class cnki
Ruijie(config-url-group)#relate user-group cnki
Ruijie(config-url-group)#exit
Ruijie(config)#
```

(3). Enable the DNS parser on the interface.

```
Ruijie(config)#interface tenGigabitEthernet 1/1/1
Ruijie(config-if-TenGigabitEthernet 1/1/1)#dns-parser enable
Ruijie(config-if-TenGigabitEthernet 1/1/1)#exit
Ruijie(config)#
```

(4). Configure PBR.

The gateway address of China Telecom 1 is 177.136.69.1.

```
Ruijie(config)# ip access-list extended 100
Ruijie(config-ext-nacl)# permit ip any user-group cnki
Ruijie(config-ext-nacl)#exit
Ruijie(config)#
Ruijie(config)# route-map cnki_policy
Ruijie(config-route-map)# match ip address 100
Ruijie(config-route-map)# set ip next-hop 177.136.69.1
Ruijie(config-route-map)#exit
Ruijie(config)#
```

Configuring DDNS

Overview

Dynamic Domain Name Server (DDNS) is a method of automatically updating a name server in the Domain Name System (DNS), often in real time, with the active DDNS configuration of its configured hostnames, addresses or other information.

Configuring DDNS

Default Configuration

Feature	Default Settings
Oray client	Disabled by default.

Configuring Oray Client

Command	Description
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **peanut username** *test* **password** *test*	Configures the username and password for the oray client.
Ruijie(config)#**end**	Returns the privileged EXEC mode.
Ruijie#**show running**	Verifies the configuration.

To remove the oray client configuration, run the **no** form of this command.

The following example configures an oray client.

```
Ruijie# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# peanut username test password test
Ruijie(config)#
```

Configuring Routing Protocol

Configuring Protocol-Independent

Configuring the IP Routing

Configuring Static Routes

Static routes are manually configured to send the packets to the specified target network. It is essential to configure the static routes when the routes of some target network cannot be learned by the dynamic routing protocols. Usually, a default static route is configured for the packets without the routes.

To configure static routes, execute the following commands in global configuration mode:

Command	Function
Ruijie(config)# **ip route** [**vrf** *vrf_name1*] *network mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [*distance*] [**tag** *tag*] [**permanent** | **track** *object-number*] [**weight** *weight*]	Configures the static routes. The vrf vrf-name1 parameter can be used to specify the VRF for the routes.
Ruijie(config)# **no ip route** [**vrf** *vrf_name1*] *network mask*	Deletes the static routes.
Ruijie(config)# **ip static route-limit** *number*	Specifies the upper limit of the static routes.
Ruijie(config)# **no ip static route-limit**	Restores the static routes to the default maximum values.

For examples of configuring the static routes, see the "Example of Replacing the Static Routes with the Dynamic Routes" section.

If the static routes are not deleted, they will be retained on Ruijie products permanently. However, you can replace the static routes with the better routes learned by the dynamic routing protocols. Better routes mean that they have smaller management distances. All routes, including the static ones, carry the parameters of the management distance. The following table shows the management distances for various route sources retained on Ruijie products:

Route source	Default management distance
Directly connected network	0
Static route	1
OSPF route	110
ISIS route	115
RIP route	120
Unreachable route	255

Note

The static route redistribution shall be configured if the static routes require to be advertised by the dynamic routing protocols such as RIP and OSPF.

When an interface is "down", all routes to the interface disappear from the routing table. In addition, when Ruijie products fail to find the forwarding route to the next-hop address for a static route, the static route will also disappear from the routing table.

When the specified VRF static routes are added to the corresponding VRF, if the egress is specified at the same time, the addition fails when the VRF of the egress does not match the specified VRF. If no VRF is specified, it is added to the global routing table by default.

If the specified VRF is a multi-protocol VRF, the static route can be configured only for the multi-protocol VRF that is configured for the IPv4 address family. When the IPv4 address family of the VRF is deleted, the IPv4 static route of the VRF will also be deleted.

If the association of the static route with the track object is specified, and if the track object is advertised to be inactive, the static route also takes no effect.

By default, the weight of the static route is 1. To view the static route of non-default weights, execute the command **show ip route weight**. The weight parameter is used to enable the WCMP function. When there are load-balanced routes to a destination address, the switch assigns data flows by their weights. The higher the weight of a route is, the more data packets the route carries. The WCMP limit is generally 32 for routers. However, the WCMP limit varies depending on switch models because their chipsets support different weights. For the detailed information about the route weight value of specific models, see the product specifications. When the sum of the load-balanced route weights exceeds the WCMP limit, the excessive routes will not take effect. For example, if the WCMP limit on a device is 8, only one of the following static route configurations takes effect:

```
Ruijie(config)#ip route 10.0.0.0 255.0.0.0 172.0.1.2 weight 6
Ruijie(config)#ip route 10.0.0.0 255.0.0.0 172.0.1.4 weight 6
Ruijie(config)#show ip route 10.0.0.0

Routing entry for 10.0.0.0/8
  Distance 1, metric 0
  Routing Descriptor Blocks:
  *172.0.1.2, generated by "static"
Ruijie(config)#show ip route weight

------------[distance/metric/weight]-----------
S    10.0.0.0/8 [1/0/6] via 172.0.1.2
```

The maximum number of the static routes is 1024 by default. If the number of the configured static routes exceeds the specified upper limit, they are not be automatically deleted, but the addition fails.

To view the configurations of the IP routing, execute the **show ip route** command to view the IP routing table. For details, see Protocol-independent Command Configuration.

Configuring the Default Route

Not all devices have a complete network-wide routing table. To allow every device to route and forward all packets, it is a common practice that the powerful core device on the network is provided with a complete routing table, while the other devices have a default route to this core device. Default routes can be transmitted by the dynamic routing protocols, and can also be manually configured on every router.

Default routes can be generated in two ways: 1) manually configuring a default static route. For details, see the "Configuring Static Routes" section; 2) manually configuring a default network.

Most internal gateway routing protocols have a mechanism that transmits the default route to the entire routing domain. The device that transmits the default route must have a default route. The transmission of the default route described in this section applies only to the RIP routing protocol. The RIP always notifies the 0.0.0.0/0 network as the default route to the RIP routing domain. For details about how the OSPF routing protocols generate and transmit the default route, see related sections in Guide on Configuring the OSPF Routing Protocols.

For generating the default static route, execute the following commands in global configuration mode:

Command	Function
Ruijie(config)# **ip default-network** network	Configures the default network.
Ruijie(config)# **no ip default-network** network	Deletes the default network.

Note

Generating the default route by using the **default-network** command must meet the following requirement: The default network is not a directly-connected interface network, but is reachable in the routing table.

Note

Under the same condition, the RIP can also transmit the default route. Alternatively, the RIP can use another way to transmit the default route, that is, by configuring the default static route or learning the 0.0.0.0/0 route by other routing protocols.

If the router has a default route, whether it is learned by the dynamic routing protocol or manually configured, when you execute the show ip route command, the "gateway of last resort" area in the routing table will show information about the last gateway. A routing table may have multiple routes as alternative default routes, but only the best default route is presented in the "gateway of last resort" area.

Configuring the Number of Equivalent Routes

To enable the load-balancing function, configure the number of the equivalent routes for control. An equivalent route is an alternative path to the same destination address. When there is only one equivalent route, one destination address can be configured with only one route, then the load-balancing function is disabled.

To configure the number of the equivalent routes, execute the following command in global configuration mode. Use the no form of this command to restore the default number of the equivalent routes.

This command is valid for both the IPv4 and the IPv6. That is to say, after configuring this command, the maximum numbers of the equivalent paths to the IPv4 and IPv6 destinations are the same as the configured value.

Command	Function

Ruijie(config)# **maximum-paths** *number*	Configures the number of the equivalent routes. The maximum number of the equivalent routes configured on different products varies. The maximum number of the equivalent routes configured on routers is 32. However, this number varies depending on the chipsets for switches. Therefore, see the prompts during the configuration.

Configuring the Route-Map

The route-map is a collection of filter policies that are independent from the detailed routing protocols and used in the routing protocols and the policy-based routing. The route-map is used to filter and modify the routing information in the routing protocols, and control the packet forwarding in the policy-based routing.

To define the route-map, execute the following commands in global configuration mode:

Command	Function
Ruijie(config)# **route-map** *route-map-name* [[**permit** | **deny**] *sequence*]	Defines the route-map.
Ruijie(config)# **no route-map** *route-map-name* [{**permit** | **deny**} *sequence*]	Deletes the route-map.

When configuring the rules for a route-map, you can execute one or more **match** or **set** commands. If there is no **match** command, all routes are matched. If there is no **set** command, no operation is performed.

To define the matching conditions for the rules, execute the following commands in route-map configuration mode:

Command	Function
Ruijie(config-route-map)# **match community** {*standard-list-number* | *expanded-list-number* | *community-list-name*} [**exac-match**]…	Matches the community attribute of the BGP route.
Ruijie(config-route-map)# **match interface** [*interface-type interface-number*…]	Matches the next-hop interface of the route.
Ruijie(config-route-map)# **match ip address** *access-list-number* [*access-list-number*…]	Matches the IP address in the ACL.
Ruijie(config-route-map)# **match ip next-hop** *access-list-number* [*access-list-number*…]	Matches the next-hop IP address in the ACL.
Ruijie(config-route-map)# **match ip route-source** *access-list-number* [*access-list-number*…]	Matches the route source IP address in the ACL.
Ruijie(config-route-map)# **match ipv6 address** { *access-list-name* | **prefix-list** *prefix-list-name* }	Matches the IPv6 ACL or prefix list.
Ruijie(config-route-map)# **match ipv6 next-hop** { *access-list-name* | **prefix-list** *prefix-list-name* }	Matches the next-hop IP address in the ACL or prefix list.
Ruijie(config-route-map)# **match ipv6 route-source** { *access-list-name* | **prefix-list** *prefix-list-name* }	Matches the route source IP address in the ACL or prefix list.
Ruijie(config-route-map)# **match metric** *Metric*	Matches the route metric value. The metric value is in the range from 0 to 4294967295.

Ruijie(config-route-map)# **match origin** {**egp** | **igp** | **incomplete**}	Matches the route origin type.
Ruijie(config-route-map)# **match route-type** {**local** | **internal** | {{**external** | **nssa-external**} [**type-1**][**type-2**]} | [**level-1** | **level-2**]}	Matches the route type.
Ruijie(config-route-map)# **match tag tag**	Matches the route tag value. The tag value is in the range from 0 to 4294967295.

To define operations after matching, execute the following commands in route-map configuration mode:

Command	Function
Ruijie(config-route-map)# **set aggregator as** *as-num ip_addr*	Sets the AS attribute value for the route aggregator.
Ruijie(config-route-map)# **set as-path prepend** *as-number*	Sets the AS_PATH attribute value.
Ruijie(config-route-map)# **set comm-list** *community-list-number* | *community-list-name delete*	Deletes all COMMUNITY attribute values in the COMMUNITY_LIST.
Ruijie(config-route-map)# **set community** {*community-number*[*community-number*…] **additive** | **none**}	Sets the COMMUNITY attribute value.
Ruijie(config-route-map)# **set dampening** *half-life reuse suppress max-suppress-time*	Sets the route dampening parameters.
Ruijie(config-route-map)#**set extcommunity** {**rt** *extend-community-value* | **soo** *extend-community-value*}	Sets the extended community attribute value.
Ruijie(config-route-map)# **set interface** *interface-type interface-number*	Sets the interface for forwarding packets.
Ruijie(config-route-map)# **set ip default next-hop** *ip-address* [*weight*] [*ip-address* [*weight*] …]	Sets the default next-hop IP address.
Ruijie(config-route-map)# **set ipv6 default next-hop** *global-ipv6-address* [*weight*] [*global-ipv6-address* [*weight*]...]	Sets the default next-hop IPv6 address.
Ruijie(config-route-map)# **set ip next-hop** *ip-address* [*weight*] [*ip-address* [*weight*]…]	Sets the next-hop IP address.
Ruijie(config-route-map)# **set ipv6** [**vrf** *vrf-name* | **global**] **next-hop** *global-ipv6-address* [*weight*] [*global-ipv6-address* [*weight*]]	Sets the next-hop IPv6 address. If the **vrf** *vrf-name* parameter is specified, the VRF is crossed when packets are forwarded. If the **global** parameter is specified, packets are forwarded globally from the VRF. If the [**vrf** *vrf-name* | **global**] parameter is not specified, the IPv6 packets will inherit the VRF during transmission. That is, the next hop belongs to the VRF that receives the IPv6 packets.
Ruijie(config-route-map)# **set level** {**stub-area** | **backbone** | **level-1** | **level-1-2** | **level-2**}	Sets the routing area.
Ruijie(config-route-map)# **set local-preference** *number*	Sets the LOCAL_PREFERENCE value.
Ruijie(config-route-map)# **set metric** *metric*	Sets the metric value for the redistributed route.

Ruijie(config-route-map)# **set metric** [+ *metric-value* |- *metric-value* | *metric-value*]	Sets the metric type for the redistributed route.
Ruijie(config-route-map)# **set metric-type** {**type-1** | **type-2** | **external** | **internal**}	Sets the metric type for the redistributed route.
Ruijie(config-route-map)# **set next-hop** *next-hop*	Sets the next-hop IP address for the redistributed route. next-hop: indicates the next-hop IP address.
Ruijie(config-route-map)# **set origin** {**egp** | **igp** | **incomplete**}	Sets the route origin attribute.
Ruijie(config-route-map)#**set originator-id** *ip-addr*	Sets the route originator ID.
Ruijie(config-route-map)# **set tag** *tag*	Sets the tag value for the redistributed route.
Ruijie(config-route-map)# **set weight** *number*	Sets the BGP route weight.

Whether a route-map supports the **match** command and the **set** command depends on applications associated with the route-map. The general instructions are as follows:

- When you configure commands associated with a route-map, the system displays a prompt when the configured match command or the set command is inapplicable to the current applications associated with the route-map.
- When you configure a route-map, the match command, or the set command, the system displays a prompt when any match command or set command is inapplicable to any application associated with the route-map.

For examples of displaying the prompt when the command is not applicable, see the "*Example of Route-Map Configuration*" section.

Caution The two instructions are inapplicable to the association of the policy-based routing with the route-map.

Redistributing Routes

Configuring Route Redistribution

To enable the device to run multiple routing protocol processes, Ruijie products provide the function for redistributing the route information from one routing process to another routing process. For example, you can redistribute the routes in the OSPF routing area to the RIP routing area, and vice versa. All IP routing protocols support the mutual route redistribution.

For route redistribution, route-maps are usually used to control the mutual route redistribution between two routing areas

To redistribute routes from one routing area to another and control the route redistribution, execute the following commands in routing process configuration mode:

Command	Function
Ruijie(config-router)# **redistribute** *protocol* [*process-id*] [metric *metric*] [**metric-type** *metric-type*] [**match internal** | **external** *type*| **nssa-external** *type*] [[**tag** *tag*] [**route-map** *route-map-name*] [**subnets**]	Redistributes the routes. Protocol (protocol type): bgp, connected, isis, rip, static

Ruijie(config-router)# **default-metric** *metric*	Sets default metric values for all redistributed routes.

The route redistribution may easily cause loops, so be careful when performing the operation.

Note

When the route redistribution is configured in the OSPF routing process, the metric value of 20 is allocated to the redistributed routes with the type of Type-2 by default. This type of the routes is the least credible routes to the OSPF.

Configuring Default Route Distribution

To advertise the default route, it is necessary for the routing protocol to introduce the default route to the process, or enforce to generate a default route.

To configure the default route distribution, execute the following commands in routing process configuration mode:

Command	Function
Ruijie(config-router)# **default-information originate** [**always**] [**metric** *metric*] [**metric-type** *type*] [**route-map** *map-name*]	Introduces the default route to the routing protocol process and advertises the default route. **always** (optional): always introduces a default route to the process no matter whether the default route exists in the local routing table. **metric** (optional): sets the metric value for the introduced default route. **metric-type** (optional): sets the metric type for the introduced default route. **route-map** (optional): filters and sets the introduced default route.
Ruijie(config-router)# no default-information originate [always] [metric metric] [metric-type type] [route-map map-name]	Cancels the operation for introducing the default route to the routing protocol process and advertising the default route.

Configuring Route Filtering

The route filtering is the process to control the inbound/outbound routes so that the device only learns the necessary and predictable routes, and only advertises the necessary and predictable routes to external trusted devices. The divulgence and chaos of the routes may affect the running of the network. Therefore, it is essential to configure the route filtering, especially for telecom operators and on financial service networks**.**

Controlling Route Updating Advertising

To prevent other routers on a local network from learning unnecessary information, you can control the route updating advertisement to prevent the specified route from updating.

To prevent the route updating advertisement, execute the following commands in routing process configuration mode:

Command	Function
Ruijie(config-router)# **distribute-list** {[*access-list-number* | *access-list-name*] **prefix** *prefix-list-name*} **out** [*interface-type interface-number*| *protocol*]	According to the ACL rules, permits or denies some routes. **prefix**: This keyword specifies the prefix list for filtering the routes. The prefix list should be separately configured by using the ip prefix-list command.
Ruijie(config-router)# **no distribute-list** {[*access-list-number* | *access-list-name*] **prefix** *prefix-list-name*} **out** [*interface-type interface-number*| *protocol*]	Cancels the operation for preventing the route updating advertising.

When you configure the OSPF, you cannot specify the interface. This feature is only applicable to the external routes in the OSPF routing area.

Controlling the Process of Route Updating

To avoid processing some specified routes of the inbound route updating packets, you can configure this feature, which does not apply to the OSPF routing protocols.

To control the route updating processing, execute the following commands in routing process configuration mode:

Command	Function
Ruijie(config-router)# **distribute-list** {[*access-list-number* | *access-list-name*] | **prefix** *prefix-list-name* [**gateway** *prefix-list-name*]| **gateway** *prefix-list-name*} **in** [*interface-type interface-number*]	According to the ACL rules, permits or denies receiving the specified inbound routes. **prefix**: This keyword specifies the prefix list for filtering the routes. The prefix list should be separately configured by using the ip prefix-list command. **gateway**: Uses the prefix list to filter the inbound routes according to the route sources.
Ruijie(config-router)# **no distribute-list** {[*access-list-number* | *access-list-name*] | **prefix** *prefix-list-name* [**gateway** *prefix-list-name*]| **gateway** *prefix-list-name*} **in** [*interface-type interface-number*]	Cancels the operation for controlling the process of the route updating.

Configuring Fast Reroute

When a link or router fails, the packets that need to forwarded through this link or router will be lost or a loop will be generated to cause service suspension. This problem can be avoided by configuring the router with the static fast reroute function.

Please refer to *Configuring OSPF* for OSPF fast rerouting configuration details.

Run the following commands to enable a static fast reroute in global configuration mode.

Command	Function
Ruijie(config)# route-map fast-reroute	Enters the route map configuration mode.
Ruijie(config-route-map)# set fast-reroute backup-nexthop GigabitEthernet 0/1 192.168.1.2	Configures a backup interface and backup next hop of the fast reroute.
Ruijie(config-route-map)# exit	Returns to the global configuration mode.
Ruijie(config)# ip fast-reroute route-map fast-reroute	Configures a static fast reroute.

Configuring the Key Chain

The key chain is used to manage authentication keys, assign key IDs, and specify the authentication string and the lifetime in the sending and receiving directions. Each key is identified and stored using a unique ID.

To manage the authentication keys, run the following commands in global configuration mode.

Command	Function
Ruijie(config)# **key chain** *key-chain-name*	Configures the key chain.
Ruijie(config-keychain)# **key** *key-id*	Configures the key ID.
Ruijie(config-keychain-key)# **key-string** [**0**|**7**] *text*	Configures the authentication string.
Ruijie(config-keychain-key)# **accept-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*}	Configures the lifetime in the receiving direction.
Ruijie(config-keychain-key)# **send-lifetime** *start-time* {**infinite** | *end-time* | **duration** *seconds*}	Configures the lifetime in the sending direction.
Ruijie(config-keychain-key)# **end**	Exits key chain configuration mode.
Ruijie# **show key chain**	Shows configuration information about key chains.

Configuration Examples

Example of Route-Map Configuration

The route-map can be configured very flexibly and applies to the route redistribution and the policy-based routing configuration. No matter how the route-map is used, the configuration principle is the same, except that different command sets are used. Even if the route-map is used for the route redistribution, different routing protocols can use different commands.

The following examples show that the RIP routes are redistributed based on the OSPF routing protocols. It is required that only the RIP routes whose hops are 4 be redistributed. The type of these routes is the external route type-1, the default metric value is 40, and the route tag is set to 40 in the OSPF routing area.

# Configure the OSPF.

```
Ruijie(config)# router ospf 1
Ruijie(config-router)# redistribute rip subnets route-map redrip
Ruijie(config-router)# network 192.168.12.0 0.0.0.255 area 0
```

# Configure the access control list.

```
Ruijie(config)# access-list 10 permit 200.168.23.0 0.0.0.255
```

# Configure the route-map.

```
Ruijie(config)# route-map redrip permit 10
Ruijie(config-route-map)# match metric 4
Ruijie(config-route-map)# set metric 40
Ruijie(config-route-map)# set metric-type type-1
Ruijie(config-route-map)# set tag 40
```

The following examples show that the OSPF routes are redistributed based on the RIP routing protocols. It is required that only the OSPF routes whose tag is 10 be redistributed. The default metric value of these routes is set to 10.

# Configure the RIP.

```
Ruijie(config)# router rip
Ruijie(config-router)# version 2
Ruijie(config-router)# redistribute ospf 1 route-map redospf
Ruijie(config-router)# network 200.168.23.0
```

# Configure the route-map.

```
Ruijie(config)# route-map redospf permit 10
Ruijie(config-route-map)# match tag 10
Ruijie(config-route-map)# set metric 10
```

The following examples show that the OSPF routes are redistributed based on the RIP routing protocols. Since rules that are not supported in the route-map application have been configured, when the route redistribution is associated with the route-map, a prompt displays indicating that the application does not support the rules.

# Configure the route-map.

```
Ruijie(config)# route-map redrip permit 10
Ruijie(config-route-map)# match length 1 3
Ruijie(config-route-map)# match route-type external
Ruijie(config-route-map)# set level backbone
```

# Configure the OSPF.

```
Ruijie(config)# router ospf 1
Ruijie(config-router)# redistribute rip subnets route-map redrip
% ospf redistribute rip not support match length
% ospf redistribute rip not support match route-type
% ospf redistribute rip not support set level backbone
```

Example of the Static Route Redistribution

- Configuration Requirements

A device exchanges route information with other devices through the RIP. In addition, there are three static routes that require the route redistribution based on the RIP. The RIP only allows to advertise two routes 172.16.1.0/24 and 192.168.1.0/24.

- Specific Configurations of the Routers

This is a common configuration example in practice for route filtering according to the distribution list. Note that the metric value is not specified for the routes to be redistributed. Since a static route is redistributed, the RIP automatically assigns the metric value. During the RIP configuration, the version must be specified and the route aggregation must be disabled because the access list allows the 172.16.1.0/24 route. To advertise the route externally, the RIP protocol must first support the classless route, and the route cannot be aggregated to the 172.16.0.0/16 network.

# Configure the static route.

```
Ruijie(config)# ip route 172.16.1.0 255.255.255.0 172.200.1.2
Ruijie(config)# ip route 192.168.1.0 255.255.255.0 172.200.1.2
Ruijie(config)# ip route 192.168.2.0 255.255.255.0 172.200.1.4
```

# Configure the RIP.

```
Ruijie(config)# router rip
Ruijie(config-router)# version 2
Ruijie(config-router)# redistribute static
Ruijie(config-router)# network 192.168.34.0
Ruijie(config-router)# distribute-list 10 out static
Ruijie(config-router)# no auto-summary
```

# Configure the extended ACL.

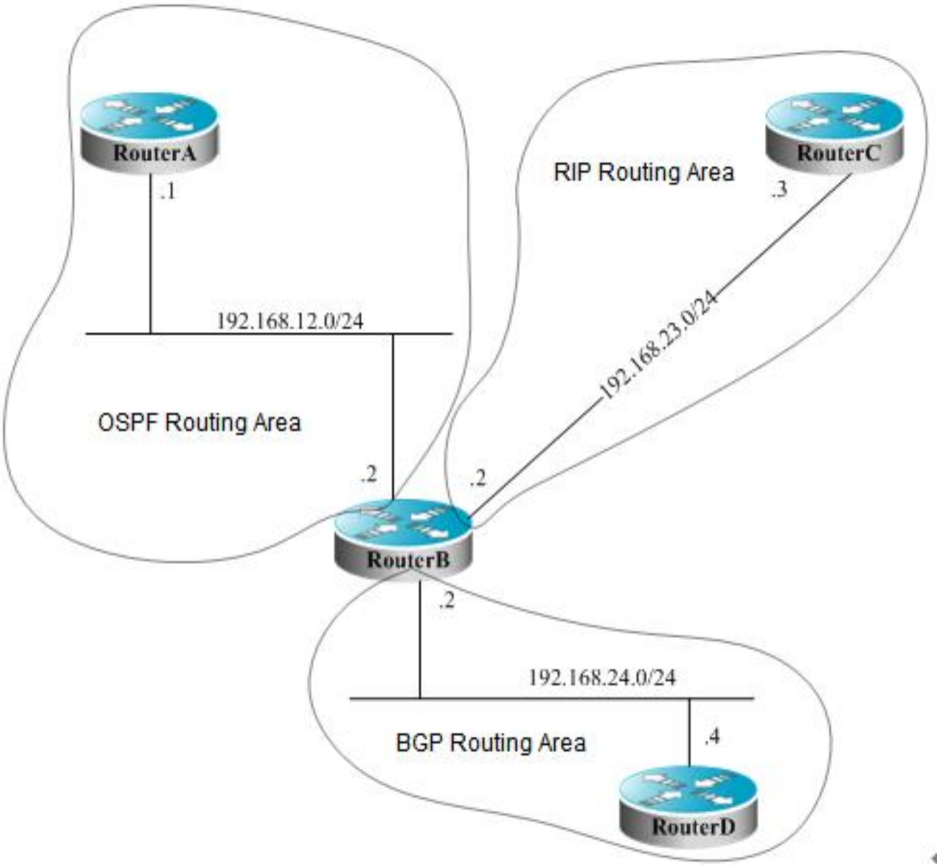
```
Ruijie(config)# ip access-list extended EXT_ACL
Ruijie(config-ext-nacl)#10 permit ip 192.168.1.0 0.0.0.255
any
Ruijie(config-ext-nacl)#10 permit ip 172.16.1.0 0.0.0.255 any
```

Example of Dynamic Routing Protocol Redistribution

- Configuration Requirements

The connection among four routers is shown in Figure 1. Router A belongs to the OSPF routing area. Router C belongs to the RIP routing area. Router D belongs to the BGP routing area. Router B is connected to the three routing areas. Router A advertises the two routes 192.168.10.0/24 and 192.168.100.1/32. Router C advertises the two routes 200.168.3.0/24 and 200.168.30.0/24. Router D advertises the two routes 192.168.4.0/24 and 192.168.40.0/24.

Figure 1 Dynamic routing protocol redistribution



On Router B, the OSPF redistributes the RIP routes with the route type of Type-1, redistributes the BGP routes whose community attribute is 11:11 in the BGP routing area. The RIP redistributes the 192.168.10.0/24 route whose metric value is set to 3 in the OSPF routing area, and advertises a default route to the RIP routing area.

- Specific Configurations of the Routers

When the routing protocols redistribute the routes among each other, the simple route filtering can be controlled by using the distribution list. However, different attributes must be set for different routes, which cannot be implemented by using the distribution list. In this case, a route-map must be used for control. The route-map provides more control functions than the distribution list, but the router configuration is more complex. Therefore, do not use the route-map if possible. The following examples use the route-map to match the community attribute of the BGP routes.

Configurations of Router A:

# Configure the network interface.

```
Ruijie(config)# interface gigabitEthernet 0/0
Ruijie(config-if)# ip address 192.168.10.1 255.255.255.0
Ruijie(config)# interface loopback 1
Ruijie(config-if)# ip address 192.168.100.1 255.255.255.255
Ruijie(config-if)# no ip directed-broadcast
Ruijie(config)# interface gigabitEthernet 0/1
Ruijie(config-if)# ip address 192.168.12.1 255.255.255.0
```

# Configure the OSPF.

```
Ruijie(config)# router ospf 12
Ruijie(config-router)# network 192.168.10.0 0.0.0.255 area 0
Ruijie(config-router)# network 192.168.12.0 0.0.0.255 area 0
Ruijie(config-router)# network 192.168.100.0 0.0.0.255 area 0
```

Configurations of Router B:

# Configure the network interface.

```
Ruijie(config)# interface gigabitEthernet 0/0
Ruijie(config-if)# ip address 192.168.12.2 255.255.255.0
Ruijie(config)# interface Serial 1/0
Ruijie(config-if)# ip address 192.168.23.2 255.255.255.0
Ruijie(config)# interface gigabitEthernet 0/1
Ruijie(config-if)# ip address 192.168.24.2 255.255.255.0
```

#Configure the OSPF and specify the redistribution route type.

```
Ruijie(config)# router ospf 12
Ruijie(config-router)# redistribute rip metric 100 metric-type 1 subnets
Ruijie(config-router)# redistribute bgp route-map ospfrm subnets
Ruijie(config-router)# network 192.168.12.0 0.0.0.255 area 0
```

#Configure the RIP and use the distribution list to filter the redistributed routes.

```
Ruijie(config)# router rip
Ruijie(config-router)# redistribute ospf 12 metric 3
Ruijie(config-router)# network 192.168.23.0
Ruijie(config-router)# distribute-list 10 out ospf
Ruijie(config-router)# default-information originate always
Ruijie(config-router)# no auto-summary
```

# Configure the BGP.

```
Ruijie(config)# router bgp 2
Ruijie(config-router)# neighbor 192.168.24.4 remote-as 4
Ruijie(config-router)# address-family ipv4
Ruijie(config-router-af)# neighbor 192.168.24.4 activate
Ruijie(config-router-af)# neighbor 192.168.24.4 send-community
```

# Configure the route-map.

```
Ruijie(config)# route-map ospfrm
Ruijie(config-route-map)# match community cl_110
```

# Define the access list.

```
Ruijie(config)# access-list 10 permit 192.168.10.0
```

# Define the community list.

```
Ruijie(config)# ip community-list standard cl_110 permit 11:11
```

Configurations of Router C:

# Configure the network interface.

```
Ruijie(config)# interface gigabitEthernet 0/0
Ruijie(config-if)# ip address 192.168.30.1 255.255.255.0
Ruijie(config)# interface gigabitEthernet 0/1
Ruijie(config-if)# ip address 192.168.3.1 255.255.255.0
Ruijie(config)# interface serial 1/0
Ruijie(config-if)# ip address 192.168.23.3 255.255.255.0
```

# Configure the RIP.

```
Ruijie(config)# router rip
Ruijie(config-router)# network 192.168.23.0
Ruijie(config-router)# network 192.168.3.0
Ruijie(config-router)# network 192.168.30.0
```

Configurations of Router D:

# Configure the network interface.

```
Ruijie(config)# interface gigabitEthernet 0/0
Ruijie(config-if)# ip address 192.168.40.1 255.255.255.0
Ruijie(config)# interface gigabitEthernet 0/1
Ruijie(config-if)# ip address 192.168.4.1 255.255.255.0
Ruijie(config)# interface serial 1/0
Ruijie(config-if)# ip address 192.168.24.4 255.255.255.0
```

# Configure the BGP.

```
Ruijie(config)# router bgp 4
Ruijie(config-router)# neighbor 192.168.24.2 remote-as 2
Ruijie(config-router)# redistribute connected route-map bgprm
Ruijie(config-router)# address-family ipv4
Ruijie(config-router-af)# neighbor 192.168.24.2 activate
Ruijie(config-router-af)# neighbor 192.168.24.2 send-community
```

# Configure the route-map.

```
Ruijie(config)# route-map bgprm
Ruijie(config-route-map)# match community 22:22
```

The OSPF routes found on Router A:

```
O E1 192.168.30.0/24 [110/101] via 192.168.12.2, 00:04:07, FastEthernet0/1
O E1 192.168.3.0/24 [110/101] via 192.168.12.2, 00:04:07, FastEthernet0/1
```

The RIP routes found on Router C:

```
R    0.0.0.0/0 [120/1] via 192.168.23.2, 00:00:00, Serial1/0
R    192.168.10.0/24 [120/2] via 192.168.23.2, 00:00:00, Serial1/0
```

Example of Fast Reroute Configuration

- Configuration Requirements

Three routers A, B and C are connected with each other via static routes. It is required that when the link between B and C fails, the service will be fast removed to the link between A and C and reach B after passing through C and A.

- Specific Configurations

Router A:

# Configure an Ethernet interface

```
Ruijie(config)interface GigabitEthernet 0/1
Ruijie(config-if)ip address 192.168.1.1 255.255.255.0
Ruijie(config-if)bfd interval 200 min_rx 200 multiplier 5
Ruijie(config)interface GigabitEthernet 0/2
Ruijie(config-if)ip address 192.168.2.1 255.255.255.0
Ruijie(config-if)bfd interval 200 min_rx 200 multiplier 5
```

# Configure a static route

```
Ruijie(config)ip route 1.1.1.1 255.255.255.255 192.168.2.2
```

Router B:

# Configure an Ethernet interface

```
Ruijie(config)interface GigabitEthernet 0/1
Ruijie(config-if)ip address 192.168.2.2 255.255.255.0
Ruijie(config-if)bfd interval 200 min_rx 200 multiplier 5
Ruijie(config)interface GigabitEthernet 0/2
Ruijie(config-if)ip address 192.168.3.1 255.255.255.0
Ruijie(config-if)bfd interval 200 min_rx 200 multiplier 5
Ruijie(config)interface loopback 1
Ruijie(config-if)ip address 1.1.1.1 255.255.255.255
```

Router C:

# Configure an Ethernet interface

```
Ruijie(config)interface GigabitEthernet 0/1
Ruijie(config-if)ip address 192.168.1.2 255.255.255.0
Ruijie(config-if)bfd interval 200 min_rx 200 multiplier 5
Ruijie(config-if)carrier-delay 0
Ruijie(config)interface GigabitEthernet 0/2
Ruijie(config-if)ip address 192.168.3.2 255.255.255.0
Ruijie(config-if)bfd interval 200 min_rx 200 multiplier 5
Ruijie(config-if)carrier-delay 0
```

# Configure a route map

```
Ruijie(config)access-list 1 permit host 1.1.1.1
Ruijie(config)route-map frr
Ruijie(config-route-map)match ip address 1
Ruijie(config-route-map)set fast-reroute backup-nexthop GigabitEthernet 0/1 192.168.1.1
```

# Configure a static route and a fast reroute

```
Ruijie(config)ip fast-reroute route-map frr
Ruijie(config)ip route 1.1.1.1 255.255.255.255 192.168.3.1
```

Configuring Policy-based Routing

Understanding Policy-based Routing

Overview

Policy-based Routing offers a more flexible packet routing forwarding mechanism than destination address-based routing forwarding, which enables you to route IPv4/IPv6 packets by elements like source address, destination address, port number and packet length.

In general, user networks apply different bandwidths from different ISPs. Meanwhile, to ensure resources for important users in the same user environment, the system needs to selectively forward packets rather than forwarding packets by the general routing table. In this case, policy-based routing takes full advantages of ISP resources and satisfy these flexible and diversified applications.

IP/IPv6 policy-based routing takes effect only on the packets received on interfaces, without any control on the packets sent from interfaces. Applying policy-based routing on an interface will check all the packets received on the interface. The packets not matching any policy of the routing map are forwarded by the general routing table, but the ones matching some policy of the routing map are forwarded by the policy.

Generally, policy-based routing takes preference over general routing and forwards IP/IPv6 packets in accord with defined policies. In other words, packets are forwarded by IP/IPv6 policy-based routing. If no rule of the PBR is matched, the packets are forwarded by general routings. Certainly, users can configure policy-based routing with the priority lower than general routing. Namely, the packets received on an interface are forwarded by general routing or policy-based routing in cast of no matching.

Users can configure forwarding mode like load balance or redundant backup according to real circumstances. Load balance or redundant backup is enabled on more than one next hop. The proportion of load balance can be also set. In redundant backup mode, multiple next hops are applied, that is the previous next hop has priority to take effect and the latter one takes effect only when the previous one fails. You can configure multiple next hops at the same time.

Policy-based routing falls into two types:

Policy-based routing enabled for the IP packets received on an interface. It performs PBR only on packets received on the interface instead of controlling the packets sending from the interface.

Policy-based routing enabled for the IP packets that the local device sends out. It controls IP packets sent from the local device to other devices, not the packets sent from external devices to the local device.

Basic Concepts and Features

Application Process

To use the policy-based routing, you must create a routing map for it and then apply the routing map on the interface. A routing map consists of many policies with corresponding sequence. Smaller sequence means higher priority.

Each policy consists of one or more match statements and corresponding one or more set statements. The match statement defines the matching rule of IP/IPv6 packets, and the set statement defines the processing rules of matched

IP/IPv6 packets. In the course of policy-based routing, packets are matched by priorities in descending order. Once a policy is matched, the system performs corresponding actions and quits policy-based routing.

Policy-based routing for IPv4 packets uses standard or extended ACL as matching rule. Policy-based routing for IPv6 packets, however, uses extended ACL as matching rule. For IPv6 packets, only one match ipv6 address can be configured for a policy at most.

Routing Map Policy Matching Mode

When you configure the routing map, you can specify the match mode of a policy as permit or deny, which is described as below:

- Permit: Specify the matching mode as permit, that is to apply the corresponding set rule to the IPv4/v6 packets meeting the match rules of the policy. If no match rule is met, the system applies the next policy to packets.
- Deny: Specify the matching mode as deny, that is if IPv4/v6 packets meet all match statements; the system performs common routing rather than policy-based routing.

IP/IPv6 packets are matched by the priority of every policy of the routing map in descending order. Once a policy is matched, the system performs corresponding actions and quits policy-based routing. If the packets do not match any policy of the routing map, the system performs common routing.

Next Hop Rules

Policy-based routing offers two forwarding rules-set {ip | ipv6} next-hop and set {ip | ipv6} default next-hop, which set the next hop and the egress, respectively. These two rules are described as follows:

- set {ip | ipv6} next-hop: Configure the policy-based routing’s next hop IPv4/IPv6 address, which takes precedence over common routes. The IPv4/v6 packets meeting the match rule received on the interface are first forwarded to the next hop specified by the set {ip | ipv6} next-hop command, no matter whether the real routing of the packets in the routing table and the next hop specified by the policy-based route is valid or not.
- set {ip | ipv6} default next-hop: The policy-based routing specified by this command is of the priority lower than common routes but higher than default route. For the packets meeting the match rule received on the interface, if routing in the routing table is failed or the default route is used, these packets will be forwarded to the next hop specified by this command.

The next hops specified by these two rules must be direct or otherwise the configuration does not take effect.

The priority is subject to the order of set {ip | ipv6} next-hop > network route/host route > set {ip | ipv6} default next-hop > default route. These two commands can be configured simultaneously, but only the one of higher priority takes effect.

Load Balancing Mode for Policy-based Routing Next Hop

More than one next hop can be configured in the sequence of a route map, and one of the following load balancing modes can be configured among them.

- Redundant backup: Only one next hop takes effect at a time if there are many next hops. Once the active next hop failed, another next hop will take over its works immediately.
  - When R1 of the active next hop fails, the system automatically hands over to R2 of the next next hop. When R1 recovers, the system will automatically hand over back to R1.
  - When there are many next hops in the order, for example, R1/R2/R3, R2 takes effect after you deleting and then adding R1 in the order of R2/R3/R1.

- Load balancing: Load balancing is enabled among next hops by traffic. This function is not available for the next hop in egress type.

Caution

1 Only one route map can be configured on a port. Configuring route maps repeatedly on a port will overlap the previous configurations, namely that the latest configuration takes effect.

Caution

2 Only one IPv6 ACL can be configured in the sequence of a route map in sub route map.

Caution

3 If the sub route map is configured with next hop but not ACL, all packets are matched; if the sub route map is configured with ACL but not next hop, the matched packets are forwarding by common routes; if the sub route map is not configured with ACL and next hop, all packets are forwarded by common routes.

Caution

4 The deny rule of ACE forwards packets by common routes. To meet the matching rule of policy-based routing, the deny any any command matches packets starting from the next IPv6 ACL

Caution

5 Enabling PBR will apply to incoming packets at the same time. If you do not need to apply PBR to a specific incoming IPv4/v6 packet, add “deny the specific IPv4/v6 address” in the ACL manually

Caution

6.In redundant backup mode, the IP packets matching the policy of the sub route map are forwarded to the next hop firstly resolved in the sequence. If all next hops are not resolved, the IP packets matching the policy are discarded. If the first next hop is resolved later, the IP packets matching the policy are forwarded to the first next hop.

Note

For details on the next hop of PBR set actions, refer to Rns&track Configuration Guide and Rns&track Command Reference or Link Detection Configuration Guide and DLDP Command Reference (for routers). IPv6 PBR is not supported at present

Note

For linkup of PBR and BFD, refer to BFD Configuration Guide and BFD Command Reference

Enabling Track Function

Track function can increase the insight of policy-based routing in the change of networks. When the device perceives that the next hop for forwarding failed, policy-based routing will rapidly hand the traffic over to the next valid next hop (in redundant backup mode) or all other valid next hops (in load balancing mode).

For track configuration, refer to Rns&track Configuration Guide. IPv6 PBR does not support linkup with track.

Enabling BFD Function

Linkup between policy-based routing and BFD avoids setting the policy-based routing as forwarding path when it is not reachable. If the backup forwarding path is available, the system rapidly hands over to this path.

VRF Selection Using Policy-based Routing

The PBR implementation of the VRF selection feature allows the ports that apply PBR to filter the packets based on the matching rule. If the packet matches this rule, route selection is performed in the specified VRF. Matching rule is defined in an IP access list or based on packet length. Users can balance traffic on different VRF instances as required.

In general, the packets received on an interface of a VRF are routed and forwarded through this VRF. The packets received on an interface of the global routing table are routed and forward through the global routing table. VRF selection using policy based routing can remove this limit. This feature supports VRF successor route, the route across VRFs and the route from VRF to the global routing table. In VRF successor route mode, the packets received on an interface of a VRF are routed and forwarded by the routing table of this VRF. In route across VRFs mode, the packets received on an interface of a VRF are routed and forwarded by the routing table of another specified VRF. In route from VRF to the global routing table mode, the packets received on an interface of a VRF are routed and forwarded by the global routing table.

Version 10.4(3) introduces multi-protocol VRF, which supports VRF selection using IPv6 PBR. If a single-protocol IPv4 VRF is specified, it does not take effect on IPv6 PBR. When a multi-protocol VRF is specified, if it does not configure the IPv4 address family, the multi-protocol VRF does not take effect on IPv4 PBR. Similarly, if the multi-protocol VRF does not configure the IPv6 address family, it does not take effect on IPv6 PBR. If the multi-protocol VRF configures IPv4 and IPv6 address families concurrently, the rules of the set vrf command take effect on both IPv4 and IPv6 PBRs.

Working Principles

For policy-based routing, first of all, you need to define a route map used to specify the policy of packet forwarding. The route map consists of a set of statements with permit or deny action.

Secondly, define a set of set statements in the route map to forward and control packets in order. Each statement does not refer to the previous or latter statements.

Finally, apply the policy-based routing at the inbound direction. If the policy-based routing is applied at the outbound direction, packets are forwarded by common routes.

For routers, outgoing packets can be processed by the specific policy-based routing, not the common routing table.

Protocol Specifications

None

Default Configurations

The default configurations of policy-based routing are described as follows:

Function	Default value
Load balance of many next hops	Redundance (redundant backup mode)
Next hop WCMP weight	1

Configuring Policy-based Routing

The following sections configure the basic functions of IP/IPV6 PBR.

Configuring IPv4 Policy-based Routing

You must specify a route map for the policy-based routing and create the route map before applying the policy-based routing. A route map consists of many policies with corresponding sequences. The smaller the sequence, the higher the priority is. Each policy consists of one or more match statements and corresponding one or more set statements. The match statement defines the matching rule of IPv4/IPv6 packets, and the set statement defines the processing rules of matched IPv4/IPv6 packets. In the course of policy-based routing, packets are matched by priorities in descending order. Once a policy is matched, the system performs corresponding actions and quits policy-based routing.

There are two kinds of match statements; match length and match ip address. The former statement matches packets by packet length and the latter matches packets by ACL. For a policy, you can configure only one match length statement but many match ip address statements. If both statements are configured at the same time, the action specified by the set rule of the policy is executed only when the packets match both.

Similarly, there are two types of set statements. Type 1 modifies the QoS field of IP packet, including set ip tos, set ip precedence and set ip dscp. Type 2 controls IP packet, for example, set vrf, set ip nexthop, set ip default nexthop, set interface and set default interface. Once all match rules are met, Type 1 set statements must be executed and Type 2 set statements are executed by priority in the following order:

- set vrf: Set policy-based routing as the VRF instance for IP packet routing with the priority higher than common route. The command is mutually exclusive with the set ip [default] nexthop and set [default]interface command. The IPv4 packets received on the interface that meet match rules will be routed by the routing table of the VRF instance specified by this command, no matter whether the VRF is the same as the one the interface belongs to.
- set ip nexthop: Set next hop of policy-based routing with the priority higher than common route and the one set by the set interface command. This command takes precedence over one of the following three commands. The IPv4 packets received on the interface that meet match rules will be firstly forwarded to the next hop specified by the set ip nexthop command, no matter whether the real routing of IPv4 packets in the routing table is the same as the one specified by the policy-based routing.
- set interface: Set the egress of policy-based routing with the priority higher than common route. This command takes precedence over set default interface and set ip default nexthop. The IPv4 packets received on the interface that meet match rules will be firstly forwarded through the egress specified by the set interface command, no matter whether the real routing of IPv4 packets in the routing table is the same as the egress specified by the policy-based routing.
- set default interface: Set the default interface with the priority higher than default route and the one specified by the set ip default nexthop command but lower than common route. The IPv4 packets received on the interface that meet match rules will be forwarded through the interface specified by this command in case of routing failure or the default route is used.
- set ip default nexthop: Set the policy-based routing with the priority higher than the default route but lower than common route. The IPv4 packets received on the interface that meet match rules will be forwarded to the next hop specified by this command in case of routing failure or the default route is used.

When you configure the routing map, you can specify the match mode of a policy as permit or deny, which is described as below:

- Permit: Specify the matching mode as permit, that is to apply the corresponding set rule to the IPv4/v6 packets meeting all match rules of the policy. If not all match rules are met, the system applies the next policy of the routing map to match packets.
- Deny: Specify the matching mode as permit, that is if IPv4/v6 packets meet all match statements of this packet's node, the system performs common routing rather than policy-based routing.

IPv4/IPv6 packets are matched by the priority of every policy of the routing map in descending order. Once a policy is matched, the system performs corresponding actions and quits policy-based routing. If the packets do not match any policy of the routing map, the system performs common routing.

The next hop specified by the set ip nexthop command is used for forwarding only when its tracking object is active. Track function greatly increases the insight of policy-based routing in the change of network environments, enabling PBR to adapt to dynamic changed networking topologies.

To configure a policy-based routing, perform the following steps:

1 Define an ACL as the matching rule of IP packets.

Command	**Function**
Ruijie(config)# ip access-list {extended | standard} {id | name}	Defines an ACL as the matching rule of IP packets.

2 Define a route map, which consists of many policies in sequence order. When a policy is matched, the system quits the execution of the route map.

Use the following command in global configuration mode to define a route map.

Command	**Function**
Ruijie(config)# route-map route-map-name [permit | deny] sequence	Defines a route map.
Ruijie(config)# no route-map route-map-name {[permit | deny] sequence}	Deletes a route map.

3 Define the match rule of every policy of the route map.

Use the following command in route map configuration mode to define the match rule of a policy.

Command	**Function**
Ruijie(config-route-map)# match ip address {access-list-number | access-list-name}	Matches the address in the ACL.
Or : Ruijie(config-route-map)# match length min max	Matches packet length.

4 Define the action after meeting match rule.

Use the following command in route map configuration mode to define actions after rules are matched.

Command	**Function**

Ruijie(config-route-map)# set vrf name	Routes the packets matching PBR by the routing table of the specific VRF instance.
Ruijie(config-route-map)# set ip next-hop ip-address [weight][ip-address[weight]]	Sets the next hop IP address of packets.
Ruijie(config-route-map)# set interface intf_name	Sets the egress of packets.
Ruijie(config-route-map)# set ip default next-hop ip-address[weight] [ip-address[weight]]	Sets the next hop IP address for the packets without route.
Ruijie(config-route-map)# set default interface intf_name	Sets the default egress of IP packets.
Ruijie(config-route-map)# set ip precedence	Modifies the priority of IP packet.
Ruijie(config-route-map)# set ip tos	Modifies the ToS value of IP packet.
Ruijie(config-route-map)# set ip dscp	Modifies the DSCP value of IP packet.

Caution

The set vrf, set ip [default] nexthop and set [default] interface commands cannot be configured concurrently for a policy. But the set vrf command can be configured with other set statements. The VRF must exist when you configure the VRF of policy-based routing otherwise the system prompts configuration failure.

Caution

The set ip dscp, set ip tos and set ip precedence commands cannot be configured concurrently for a policy or otherwise the corresponding domains of IP packet may be different from the expectation

Caution

The priorities of the set vrf, set ip nexthop, and set interface commands take precedence over that of common routes. IP packets matching policy-based routing are forwarded by policy-based routing, but the IP packets not matching the policy-based routing are forwarded by common routes.

Caution

The set default ip nexthop and set default interface commands are lower than common route in terms of priority. IP packets are routed and forwarded by policy-based routing only after common route failed

For details on route map configuration, refer to Protocol-independent Configuration Commands.

1 Apply the route map on the specified interface.

Use the following command in interface configuration mode to apply the policy-based routing on the specified interface.

Command	Function
Ruijie(config-if)# ip policy route-map name	Applies policy-based routing on the interface.
Ruijie(config-if)# no ip policy route-map	Removes the configuration.

2. Apply the policy-based routing to the packets sent locally.

Command	Function
Ruijie(config)# ip local policy route-map [name]	Applies the policy-based routing to the packets sent locally.
Ruijie(config)# no ip local policy route-map	Removes the configuration.

For example:

Configure policy-based routing on Fastethernet 0/0 so that all incoming packets are forwarded to the device whose next hop is 192.168.5.5.

```
Ruijie(config)# access-list 1 permit any
Ruijie(config)# route-map name
Ruijie(config-route-map)# match ip address 1
Ruijie(config-route-map)# set ip next-hop 192.168.5.5
Ruijie(config-route-map)# int fastethernet 0/0
Ruijie(config-if)# ip policy route-map name
```

3 Configure load balancing mode for policy-based routing

In redundant backup mode, the policy-based routing will automatically hand over the next valid next hop when the active next hop fails. In load balancing mode, on contrary, the traffic will be balanced on other valid next hop when the active next hop fails.

Use the following command in global configuration mode to configure load balance or redundant backup:

Command	**Function**
Ruijie(config)# ip policy {load-balance | redundance}	Configures load balance or redundant backup for policy-based routing forwarding.
Ruijie(config)# no ip policy	Removes the configuration.

Caution

In load balancing mode, Weighted Cost Multiple Path (WCMP) supports up to 4 next hops and Equal Cost Multiple Path (ECMP) supports up to 32 next hops.

Caution

In load balancing mode, Weighted Cost Multiple Path (WCMP) supports up to 4 next hops and Equal Cost Multiple Path (ECMP) supports up to 32 next hops.

Caution

For default policy-based routing, Weighted Cost Multiple Path (WCMP) supports up to 4 next hops and Equal Cost Multiple Path (ECMP) supports up to 32 next hops.

Caution

In redundant backup mode, the first resolved next hop takes effect. If all next hops are not resolved, the packets matching policy-based routing are dropped. If the originally unresolved next hop of higher priority than active next hop is resolved, the system hands over to this next hop.

Configuring IPv6 Policy-based Routing

Command	Function
Ruijie#configure terminal	Enters global configuration mode.
Ruijie(config)#ipv6 access-list access-list-name	Creates an IPv6 ACL.
Ruijie (config)#route-map route-map-name [permit | deny] sequence	Creates a route map.
Ruijie (config-route-map)#match ipv6 address access-list-name	Matches the IPv6 address in ACL.
Ruijie (config-route-map)#set ipv6 [vrf vrf-name | global] next-hop global-ipv6-address [weight][global-ipv6-address [weight]] [global-ipv6-address...]	Sets the next hop IPv6 address of packets. The [vrf vrf-name | global] parameter is supported since version 10.4(3), that is, cross VRFs and from VRF to global modes are supported. If the vrf vrf-name parameter is specified, the next hop belongs to the VRF, while if the global parameter is specified, the next hop belongs to the global. Note that the specified VRF must be a multi-protocol VRF whose IPv6 address family has been configured. VRF will be inherited when forwarding IPv6 packets if the set ipv6 next-hop command is configured. The next hop belongs to the VRF that receive the IPv6 packets and forward them internally.
Or: Ruijie (config-route-map)#set ipv6 default next-hop global-ipv6-address [weight][global-ipv6-address [weight]] [global-ipv6-address...]	Specifies the next hop IPv6 address for the packets without obvious routes in the routing table.
Ruijie (config)#interface interface-type interface-number	Enters the interface require applying PBR.
Ruijie (config-if- interface-type interface-number)#ipv6 policy route-map route-map-name	Applies policy-based routing on the interface.
Or: Ruijie (config-if- interface-type interface-number)#no pv6 policy route-map	Removes the policy-based routing applied on the interface.
Ruijie#show ipv6 policy	Shows the configuration of policy-based routing.

Or: Ruijie#show route-map	Shows the configuration of route map.

For route map configuration, refer to *Configuring Protocol-independent*.

Configuring Load Balancing Mode

Command	Function
Ruijie#configure terminal	Enters global configuration mode.
Ruijie(config)#Ipv6 policy [load-balance | redundance]	Configures load balance mode.
Ruijie(config)#no Ipv6 policy	Restores the setting to the default value.

Displaying Configuration and States

Command	Function
Ruijie#**show** { **ip** | **ipv6** } **policy**	Shows the configuration of policy-based routing.
Ruijie#**show route-map**	Shows the configuration of route map.
Ruijie#**show access-lists**	Shows the configuration of ACL.

Typical Configuration Examples

Example 1: Source address based PBR

Networking Requirements

There are two egresses of a LAN connecting to the Internet. In general, load balance and backup should be enabled for these two egresses. All streams from subnet 1 to the Internet are transmitted through GigabitEthernet 0/1 and all streams from subnet 2 to the Internet are transmitted through GigabitEthernet 0/2. If GigabitEthernet 0/1 is disconnected, the data streams on this interface should be transferred to GigabitEthernet 0/2, and vice versa.

Networking Topology

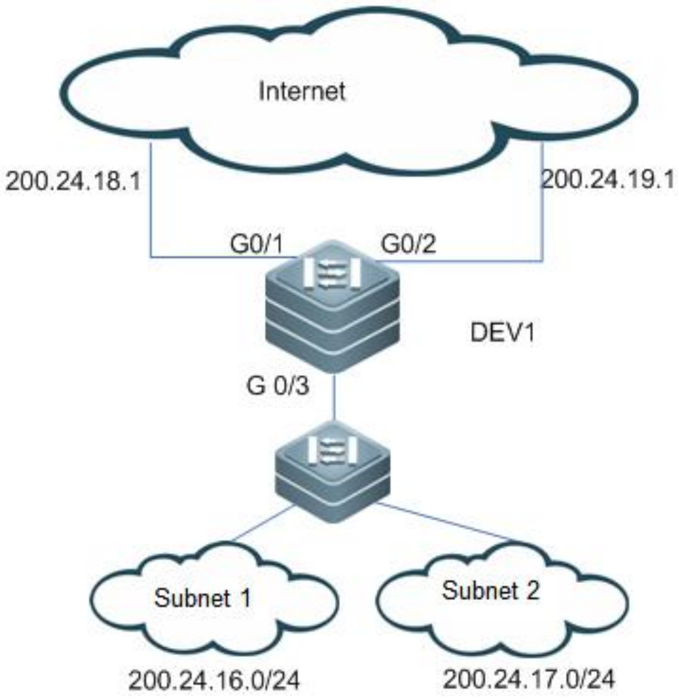


Figure 1 Network topology

As shown in the Figure-1, Layer-3 device DEV1 connects to subnets 1 and 2 through G0/3, and connects to the Internet through G0/1 and G0/2 with the next hop of 200.24.18.1 and 200.24.19.1, respectively. Subnet 1's segment is 200.24.16.0/24 and subnet 2's segment is 200.24.17.0/24.

Configuration Steps

# Create ACLs for subnet 1 and subnet 2, respectively.

```
Ruijie(config)#access-list 1 permit 200.24.16.0 0.0.0.255
Ruijie(config)#access-list 2 permit 200.24.17.0 0.0.0.255
```

# Configure a route map used to control data streams of subnet 1. Set the next hop of G0/1 prefer.

```
Ruijie(config)#route-map RM_FOR_PBR 10
Ruijie(config-route-map)#match ip address 1
Ruijie(config-route-map)#set ip nexthop 200.24.18.1
Ruijie(config-route-map)#set ip nexthop 200.24.19.1
```

# Configure a route map used to control data streams of subnet 2. Set the next hop of G0/2 prefer.

```
Ruijie(config)#route-map RM_FOR_PBR 20
Ruijie(config-route-map)#match ip address 2
Ruijie(config-route-map)#set ip nexthop 200.24.19.1
Ruijie(config-route-map)#set ip nexthop 200.24.18.1
```

# Configure redundant backup.

```
Ruijie(config)#ip policy redundance
```

# Apply policy-based routing on GigabitEthernet 0/3.

```
Ruijie(config)#interface GigabitEthernet 0/3
Ruijie(config-if)#ip policy route-map RM_FOR_PBR
```

Example 2: Enabling Track function.

Networking Requirements

There are two egresses on a LAN connecting to the Internet. In general, load balance and backup should be enabled for these two egresses. All streams from subnet 1 to the Internet are transmitted through GigabitEthernet 0/1 and all streams from subnet 2 to the Internet are transmitted through GigabitEthernet 0/2. If the next hop 200.24.18.1 fails, the data streams on this interface should be transferred to GigabitEthernet 0/2, and vice versa.

Networking Topology

As shown in Figure 1.

Configuration steps

# Track the egress GigabitEthernet 0/1’s next hop 200.24.18.1.

```
Ruijie(config)#ip rns 1
Ruijie(config-ip-rns)#icmp-echo 200.24.18.1
Ruijie(config)#track 1 rns 1
```

# Track the egress GigabitEthernet 0/2’s next hop 200.24.19.1.

```
Ruijie(config)#ip rns 2
Ruijie(config-ip-rns)#icmp-echo 200.24.19.1
Ruijie(config)#track 2 rns 2
```

# Enable the routing map to use this track object..

```
Ruijie(config)#route-map RM_FOR_PBR 10
Ruijie(config-route-map)#match ip address 1
Ruijie(config-route-map)#set ip nexthop verify-availability 200.24.18.1 track 1
Ruijie(config-route-map)#set ip nexthop verify-availability 200.24.19.1 track 2
```

# Configure a route map used to control data streams of subnet 2. Set the next hop of G0/2 prefer.

```
Ruijie(config)#route-map RM_FOR_PBR 20
Ruijie(config-route-map)#match ip address 1
Ruijie(config-route-map)#set ip nexthop verify-availability 200.24.19.1 track 2
Ruijie(config-route-map)#set ip nexthop verify-availability 200.24.18.1 track 1
```

# Configure redundant backup.

```
Ruijie(config)#ip policy redundance
```

# Apply policy-based routing on GigabitEthernet 0/3.

```
Ruijie(config)#interface GigabitEthernet 0/3
Ruijie(config-if)#ip policy route-map RM_FOR_PBR
```

Example 3: Configuring VRF Selection Using PBR

Networking Requirements

A provider edge (PE) requires applying policy-based routing to the packets received from FastEthernet 0/1. It routes the IP packets from subnet 1 by VRF1, the IP packets from subnet 2 by VRF2, and the IP packets from subnet 3 by VRF3. Other packets are routed in the public network.

Configuration Steps

# Create VRF instances.

```
Ruijie(config)#ip vrf VRF1
Ruijie(config)#ip vrf VRF2
Ruijie(config)#ip vrf VRF3
```

# Create ACLs as matching rules of the routing map.

```
Ruijie(config)#access-list 1 permit 192.168.195.0 0.0.0.255
Ruijie(config)#access-list 2 permit 192.168.196.0 0.0.0.255
Ruijie(config)#access-list 3 permit 192.168.197.0 0.0.0.255
```

# Create route maps.

```
Ruijie(config)#route-map PBR-VRF-Selection permit 10
Ruijie(config-route-map)#match ip address 1
Ruijie(config-route-map)#set vrf VRF1

Ruijie(config)#route-map PBR-VRF-Selection permit 20
Ruijie(config-route-map)#match ip address 2
Ruijie(config-route-map)#set vrf VRF2

Ruijie(config)#route-map PBR-VRF-Selection permit 30
Ruijie(config-route-map)#match ip address 3
Ruijie(config-route-map)#set vrf VRF3
```

# Import IP address of the interface to VRFs 1 to 3.

```
Ruijie(config)#interface FastEthernet 0/1
Ruijie(config-if)#ip address 192.168.195.1 255.255.255.0
Ruijie(config-if)#ip vrf receive VRF1
Ruijie(config-if)#ip vrf receive VRF2
Ruijie(config-if)#ip vrf receive VRF3
```

# Apply the policy-based routing on the interface.

```
Ruijie(config)#interface FastEthernet 0/1
Ruijie(config-if)#ip policy route-map PBR-VRF-Selection
```

The configurations are as follows if the single protocol IPv4 VRF is replaced with a multi-protocol VRF:

# Create a VRF instance.

```
Ruijie(config)#vrf definition VRF1
Ruijie(config-vrf)#address-family ipv4
Ruijie(config-vrf-af)#exit-address-family
Ruijie(config-vrf)#vrf definition VRF2
Ruijie(config-vrf)#address-family ipv4
Ruijie(config-vrf-af)#exit-address-family
Ruijie(config-vrf)#vrf definition VRF3
Ruijie(config-vrf)#address-family ipv4
Ruijie(config-vrf-af)#exit-address-family
Ruijie(config-vrf)#exit
```

# Configure an ACL as the matching rule of the routing map.

```
Ruijie(config)#access-list 1 permit 192.168.195.0 0.0.0.255
Ruijie(config)#access-list 2 permit 192.168.196.0 0.0.0.255
Ruijie(config)#access-list 3 permit 192.168.197.0 0.0.0.255
```

# Configure the routing map.

```
Ruijie(config)#route-map PBR-VRF-Selection permit 10
Ruijie(config-route-map)#match ip address 1
Ruijie(config-route-map)#set vrf VRF1

Ruijie(config)#route-map PBR-VRF-Selection permit 20
Ruijie(config-route-map)#match ip address 2
Ruijie(config-route-map)#set vrf VRF2

Ruijie(config)#route-map PBR-VRF-Selection permit 30
Ruijie(config-route-map)#match ip address 3
Ruijie(config-route-map)#set vrf VRF3
```

# Import IP address of the interface to VRFs 1 to 3.

```
Ruijie(config)#interface FastEthernet 0/1
Ruijie(config-if)#ip address 192.168.195.1 255.255.255.0
Ruijie(config-if)#vrf receive VRF1
Ruijie(config-if)#vrf receive VRF2
Ruijie(config-if)#vrf receive VRF3
```

# Apply the policy-based routing to the interface.

```
Ruijie(config)#interface FastEthernet 0/1
Ruijie(config-if)#ip policy route-map PBR-VRF-Selection
```

Example 4: Appling IPv6 Policy-based Routing on the Interface

Networking Requirements

There are two egresses on a LAN connecting to the Internet. In general, load balance and backup should be enabled for these two egresses.

Specific requirements are as follows:

- All streams from subnet 1 to the Internet are transmitted through GigabitEthernet 0/1.
- All streams from subnet 2 to the Internet are transmitted through GigabitEthernet 0/2.
- If GigabitEthernet 0/1 is disconnected, the data streams on this interface should be transferred to GigabitEthernet 0/2, and vice versa.

Networking Topology

As shown in Figure 2, Lay-3 device Device1 connects to subnets 1 and 2 through G0/3 (routed port), and connects to the Internet through G0/1 and G0/2 with the next hop of 2001::1/64 and 2002::1/64 respectively. Subnet 1's segment is 2003::/64 and subnet 2's segment is 2004::/64.

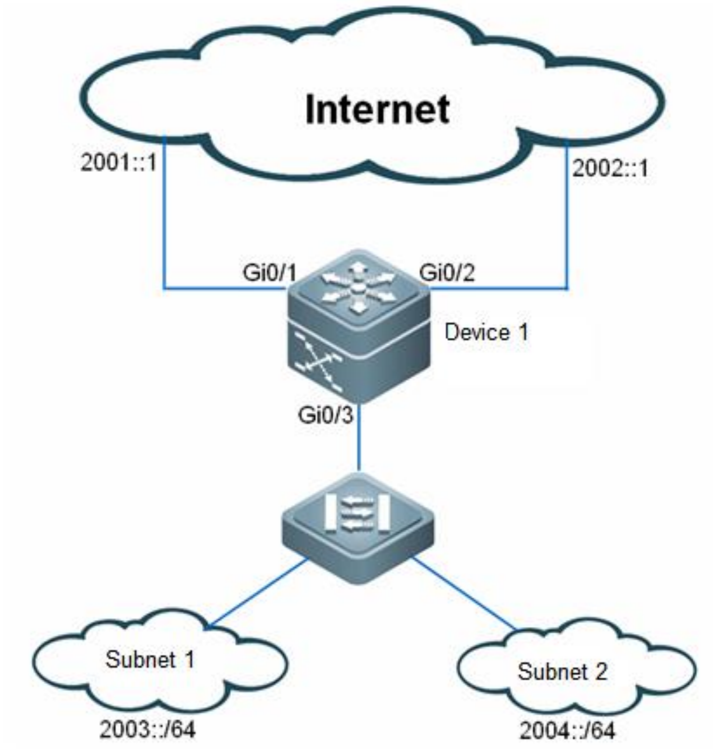


Figure 2 IPV6 PBR topology

Configuration Tips

Configuration Steps

# Create ACLs for subnet 1 and subnet 2 respectively.

```
Ruijie(config)#ipv6 access-list net1
Ruijie(config-ipv6-acl)#permit ipv6 2003::/64 any
Ruijie(config)#ipv6 access-list net2
Ruijie(config-ipv6-acl)#permit ipv6 2004::/64 any
```

# Configure a route map used to control data streams of subnet 1. Set the next hop of G0/1 prefer.

```
Ruijie(config)#route-map RM_FOR_PBR 10
Ruijie(config-route-map)#match ipv6 address net1
Ruijie(config-route-map)#set ipv6 next-hop 2001::1
Ruijie(config-route-map)#set ipv6 next-hop 2002::1
```

# Configure a route map used to control data streams of subnet 2. Set the next hop of G0/2 prefer.

```
Ruijie(config)#route-map RM_FOR_PBR 20
Ruijie(config-route-map)#match ipv6 address net2
Ruijie(config-route-map)#set ipv6 next-hop 2002::1
Ruijie(config-route-map)#set ipv6 next-hop 2001::1
```

# Configure redundant backup.

```
Ruijie(config)#ipv6 policy redundance
```

# Apply the policy-based routing on the interface GigabitEthernet 0/3.

```
Ruijie(config)#interface GigabitEthernet 0/3
Ruijie(config-if-GigabitEthernet 0/3)#ipv6 policy route-map RM_FOR_PBR
```

Verification

# Show the configuration of route map.

```
Ruijie#show route-map
route-map RM_FOR_PBR, permit, sequence 10
  Match clauses:
   ipv6 address net1
  Set clauses:
   ipv6 next-hop 2001::1 2002::1
route-map RM_FOR_PBR, permit, sequence 20
  Match clauses:
   ipv6 address net2
  Set clauses:
ipv6 next-hop 2002::1 2001::1
```

# Show the configuration of IPv6 policy-based routing.

```
Ruijie#show ipv6 policy
Interface                          Route map
GigabitEthernet 0/3                 RM_FOR_PBR
```

# Show the configuration of ACL.

```
Ruijie#show access-lists
ipv6 access-list net1
 10 permit ipv6 2003::/64 any
 (0 packets matched)
ipv6 access-list net2
 10 permit ipv6 2004::/64 any
 (0 packets matched)
```

Example 5: Configuring IPv4/IPv6 PBRs Concurrently

Networking Requirements

There are two egresses on a LAN connecting to the Internet, one of which is the egress of education network. In general, load balance and backup should be enabled for these two egresses.

Specific requirements are as follows:

- IPv4/IPv6 dual stacks are used in the networks. IPv4 and IPv6 PBRs should be enabled on an interface at the same time.
- All streams from the IPv4 education network of subnet 1 to the Internet are transmitted through the egress of education network.
- All streams from the IPv4 education network of subnet 2 to the Internet are transmitted through the egress of the Internet.
- All streams from the IPv6 education network of subnet 1 to the Internet are transmitted through GigabitEthernet 0/1.
- All streams from the IPv6 education network of subnet 2 to the Internet are transmitted through GigabitEthernet 0/2.
- Internal interactive data, for example, the data from subnet 1 to subnet 2, is transmitted using the internal dynamic route rather than policy-based routing.
- By default, data streams are transmitted through the egress of the Internet by the default route.
- If GigabitEthernet 0/1 fails, the data streams on the interface are switched over to GigabitEthernet 0/2, and vice versa.

Networking Topology

As shown in Figure 3, Device 1 connects to subnets 1 and 2 through G0/3 (routed port), and connects to the Internet through G0/1 and G0/2 with the next hop of 2001::1/64 (210.82.12.1) and 2002::1/64 (59.78.184.1) respectively. Subnet 1’s segment is 2003::/64 (202.112.144.0/25) and subnet 2’s segment is 2004::/64(218.62.95.0/24).

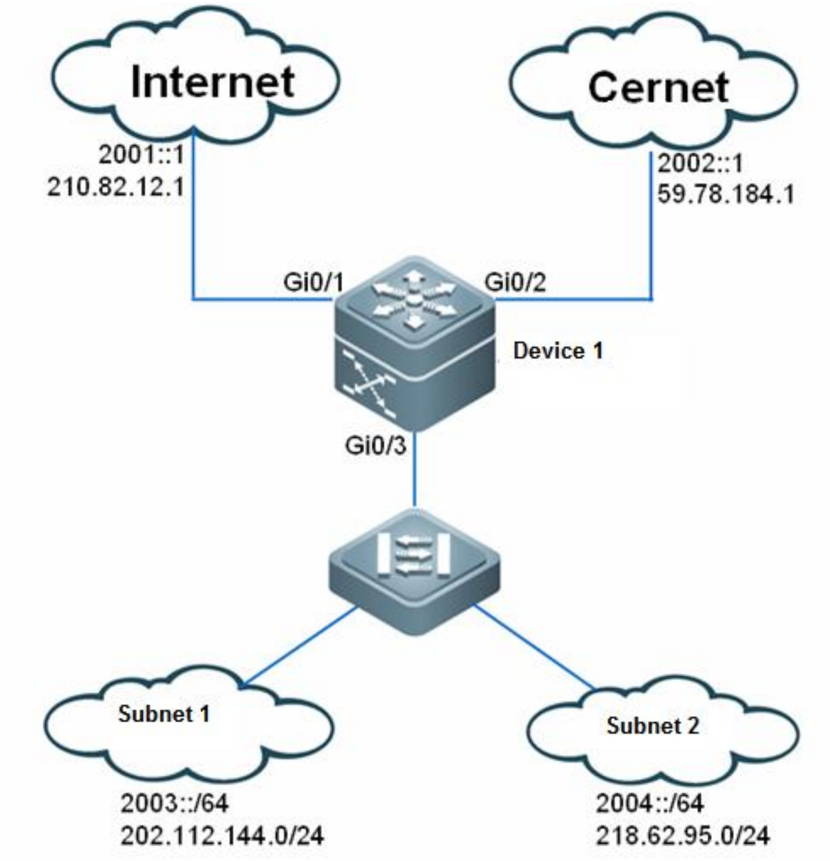


Figure 3 IPv4/IPv6 PBR topology

Configuration Tips

Configuration Steps

# Create IPv4 ACLs for subnet 1 and subnet 2, respectively.

```
Ruijie(config)#ip access-list extended 101
Ruijie(config-ip-acl)#permit ip 202.112.144.0 0.0.0.255 any
Ruijie(config)#ip access-list extended 102
Ruijie(config-ip-acl)#permit ip 218.62.95.0 0.0.0.255 any
```

# Create IPv6 ACLs for subnet 1 and subnet 2, respectively.

```
Ruijie(config)#ipv6 access-list net1
Ruijie(config-ipv6-acl)#permit ipv6 2003::/64 any
Ruijie(config)#ipv6 access-list net2
Ruijie(config-ipv6-acl)#permit ipv6 2004::/64 any
```

# Configure a route map used to control data streams of subnet 1. Set the next hop of G0/1 prefer. The default parameter is included in attributes of the IPv4 next hop.

```
Ruijie(config)#route-map RM_FOR_PBR 10
Ruijie(config-route-map)#match ip address 101
Ruijie(config-route-map)#set ip default next-hop 59.78.184.1
Ruijie(config-route-map)#set ip default next-hop 210.82.12.1

Ruijie(config-route-map)#match ipv6 address net1
Ruijie(config-route-map)#set ipv6 next-hop 2001::1
Ruijie(config-route-map)#set ipv6 next-hop 2002::1
```

# Configure a route map used to control data streams of subnet 2. Set the next hop of G0/2 prefer. The default parameter is included in attributes of the IPv4 next hop.

```
Ruijie(config)#route-map RM_FOR_PBR 20
Ruijie(config-route-map)#match ip address 102
Ruijie(config-route-map)#set ip default next-hop 210.82.12.1
Ruijie(config-route-map)#set ip default next-hop 59.78.184.1

Ruijie(config)#route-map RM_FOR_PBR 20
Ruijie(config-route-map)#match ipv6 address net2
Ruijie(config-route-map)#set ipv6 next-hop 2002::1
Ruijie(config-route-map)#set ipv6 next-hop 2001::1
```

# Configure redundant backup.

```
Ruijie(config)#ipv6 policy redundance
```

# Apply the IPv4/IPv6 policy-based routing on the interface GigabitEthernet 0/3.

```
Ruijie(config)#interface GigabitEthernet 0/3
Ruijie(config-if-GigabitEthernet 0/3)#ip policy route-map RM_FOR_PBR
Ruijie(config-if-GigabitEthernet 0/3)#ipv6 policy route-map RM_FOR_PBR
```

Verification

# Show the configuration of the route map.

```
Ruijie#show route-map
route-map RM_FOR_PBR, permit, sequence 10
  Match clauses:
    ip address 101
    ipv6 address net1
  Set clauses:
    ipv6 next-hop 2001::1 2002::1
    ip default next-hop 59.78.184.1 210.82.12.1
route-map RM_FOR_PBR, permit, sequence 20
  Match clauses:
    ip address  102
    ipv6 address net2
  Set clauses:
    ipv6 next-hop 2002::1 2001::1
    ip default next-hop 210.82.12.1 59.78.184.1
```

# Show the application of IPv6 policy-based routing.

```
Ruijie#show ipv6 policy
Interface                          Route map
GigabitEthernet 0/3                 RM_FOR_PBR
```

# Show the application of IPv4 policy-based routing.

```
Ruijie#show ip policy
Interface                          Route map
GigabitEthernet 0/3                 RM_FOR_PBR
```

# Show the configuration of ACLs.

```
Ruijie#show access-lists
Extended IP access list 101
    10 permit ip 202.112.144.0 0.0.0.255 any
Extended IP access list 102
    10 permit ip 218.62.95.0 0.0.0.255 any
```

```
IPv6 access list net1
   permit ipv6 2003::/64 any sequence 10
IPv6 access list net2
   permit ipv6 2004::/64 any sequence 10
```

Example 6: VRF Election Using IPv6 PBR

Networking Requirements

A provider edge (PE) requires applying policy-based routing to the packets received from FastEthernet 0/1. It routes the IP packets from subnet 1 by VRF1, the IP packets from subnet 2 by VRF2, and the IP packets from subnet 3 by VRF3. Other packets are routed in the public network.

Configuration Steps

# Create a VRF instance.

```
Ruijie(config)#vrf definition VRF1
Ruijie(config-vrf)#address-family ipv6
Ruijie(config-vrf-af)#exit-address-family
Ruijie(config-vrf)#vrf definition VRF2
Ruijie(config-vrf)#address-family ipv6
Ruijie(config-vrf-af)#exit-address-family
Ruijie(config-vrf)#vrf definition VRF3
Ruijie(config-vrf)#address-family ipv6
Ruijie(config-vrf-af)#exit-address-family
Ruijie(config-vrf)#exit
```

# Configure an ACL as the matching rule of the routing map.

```
Ruijie(config)#ipv6 access-list acl1
Ruijie(config-ipv6-acl)#permit ipv6 1000::/64 any
Ruijie(config-ipv6-acl)#ipv6 access-list acl2
Ruijie(config-ipv6-acl)#permit ipv6 2000::/64 any
Ruijie(config-ipv6-acl)#ipv6 access-list acl3
Ruijie(config-ipv6-acl)#permit ipv6 3000::/64 any
```

# Configure the routing map.

```
Ruijie(config-ipv6-acl)#route-map PBR-VRF-Selection permit 10
Ruijie(config-route-map)#match ipv6 address acl1
Ruijie(config-route-map)#set vrf VRF1

Ruijie(config)#route-map PBR-VRF-Selection permit 20
Ruijie(config-route-map)#match ipv6 address acl2
Ruijie(config-route-map)#set vrf VRF2
Ruijie(config)#route-map PBR-VRF-Selection permit 30
Ruijie(config-route-map)#match ipv6 address acl3
Ruijie(config-route-map)#set vrf VRF3
```

# Import IPv6 address of the interface to VRFs 1 to 3.

```
Ruijie(config)#interface FastEthernet 0/1
Ruijie(config-if)#ipv6 address 1000::1/64
Ruijie(config-if)#vrf receive VRF1
Ruijie(config-if)#vrf receive VRF2
Ruijie(config-if)#vrf receive VRF3
```

# Apply the IPv6 PBR to the interface.

```
Ruijie(config)#interface FastEthernet 0/1
Ruijie(config-if)#ipv6 policy route-map PBR-VRF-Selection
```

Configuring RIP

Overview

The Routing Information Protocol (RIP) is a relatively old routing protocol, which is widely used in small or homogeneous networks. The RIP uses the distance-vector algorithm, and so is a distance-vector protocol. The RIPv1 is defined in RFC 1058 and the RIPv2 is defined in RFC 2453. Ruijie RGOS supports both two versions.

The RIP exchanges the routing information by using UDP packets with UDP port number 520. Usually, RIPv1 packets are broadcast packets, while RIPv2 packets are multicast packets with the multicast address of 224.0.0.9. The RIP sends an update packet at the interval of 30 seconds. If a device fails to receive the route update packets from the peer within 180 seconds, it will mark all the routes from the device unreachable. After that, the device will delete these routes from its routing table if it still fails to receive any update packets within 120s.

The RIP measures the distance to the destination in hop, known as route metric. In the RIP, zero hop exists when the device directly connects to the network. One hop exists when the destination is reachable through one device and so on. If the destination is unreachable, the hop count is 16.

The RIP-enabled device can learn the default routes from the neighbors or generate its own default route. When any of the following conditions is met, Ruijie products will introduce the default route and advertise it to its neighbor devices by using the **default-information originate** command:

- IP Default-network is configured.
- Other RIPs learn the default routes or are configured with static default routes.

The RIP will send update packets to a specified network interface. If the network is not associated with the RIP routing progress, the interface will not advertise any update packets. The RIP is available in two versions: RIPv1 and RIPv2. The RIPv2 supports plain-text authentication, MD5 cryptographic text authentication, and variable length subnet masks.

Ruijie RIP offers Split Horizon to avoid a loop.

Configuring RIP

The RIP configuration task list contains:

- Create the RIP routing process (mandatory).
- Configure RIP packets in unicast mode (optional).
- Configure Split Horizon (optional).
- Define the RIP Version (optional).
- Configure the Route Aggregation function(optional).
- Configure RIP Authentication.
- Adjust the RIP clock (optional).
- Configure the RIP Route Source Address Validation (optional).
- Control RIP interface status (optional).
- Advertise the default route through the RIP interface (optional).
- Advertise the supernet route through the RIP interface (optional).

- Configure RIP VRF (optional).
- Configure RIP BFD (optional).
- Configure RIP triggered expansion (optional).
- Configure RIP graceful restart (GR) (optional).

For the following topics, see the "IP Routing Protocol Independent Feature Configuration" chapter.

- Filter RIP route information
- Redistribute routes
- Configure default route distribution

The following table describes the default configuration of RIP.

Feature	Default Setting
Network interface	After an interface joins the RIP, it receives RIPv1 and RIPv2 packets and advertises RIPv1 packets by default. By default, when advertising RIPv1 packets: ■ The interface sends the packets in broadcast mode. ■ The interface does not advertise supernet routes. By default, when advertising RIPv2 packets: ■ The interface sends the packets in multicast mode. ■ The interface automatically converges routes into classified routes. ■ The interface advertises supernet routes. The interface enables split horizon.
RIP neighbor	Undefined
Verification of the source IP address of a packet	Enabled
Timer	By default: ■ The update time is 30 seconds. ■ The expiry time is 180 seconds. ■ The clearing time is 120 seconds.
Offset list	Undefined
Automatic convergence	Enabled
Redistribution	By default, routing redistribution is disabled. If it is enabled: ■ Redistribute OSPF, that is, redistribute all sub-type routes of this instance. ■ Redistribute ISIS, that is, redistribute level-2 sub-type routes of this instance. ■ In other cases, redistribute all routes of this type. ■ The metric value for route distribution is the default one.
Default route distribution	By default, default route distribution is disabled. If it is enabled: the metric value for route distribution is the default one.
Default metric	Redistribute metric values used by routes of other protocols. The default value is 1.
Administrative distance	120

Feature	Default Setting
RIP triggered expansion	By default, RIP triggered expansion is disabled. If it is enabled: The default interval is 5 seconds for retransmitting update request and update response packets. By default, a maximum of 36 times are allowed for retransmitting update request and update response packets.
RIP GR	By default, RIP GR is disabled. The default RIP GR period is the smaller value between twice the update time and 60 seconds.

Creating the RIP Routing Process

To run RIP for a device, create the RIP routing process and define networks associated with the RIP routing process.

Use the following commands to create the RIP routing process in global configuration mode.

Command	Function
Ruijie(config)# **router rip**	Creates the RIP routing process.
Ruijie(config-router)# **network** *network-number wildcard*	Defines associated networks.

You can configure the *network-number* and *wildcard* parameters at the same time to enable the network segments of the interface IP address within the IP address range to run RIP.

If the *wildcard* parameter is not configured, by default, RGOS will enable the network segments of the interface IP address within the classified IP address range to run RIP.

Note There are two meanings for associated networks defined by the network command:

Note The RIP only advertises the route information of associated networks.

Note The RIP only advertises and receives route update messages through the interfaces of associated networks.

Configuring RIP Packets in Unicast Mode

The RIP is usually a broadcast or multicast protocol. If the RIP route information needs to be transmitted through non-broadcast networks, a device needs to be configured to support that the RIP advertises route information update packets in unicast mode.

Use the following commands to advertise RIP information update messages in unicast mode in RIP routing process configuration mode.

Command	Function
Ruijie(conf-router)# neighbor ip-address	Configures RIP packet advertising in unicast mode.

This command enables you to control an interface about whether to advertise RIP route update packets and forbid advertising route update packets in broadcast mode through an interface. You need to configure the **passive-interface** command in routing progress configuration mode. For related descriptions on the restriction of route message advertisements, see the "Route Filtering Configuration" of *Configuring Protocol Independent*.

Note

During the configuration of FR and X.25, if the broadcast keyword is specified for IP address mapping, the neighbor command is not required because the command is mainly used for reducing broadcast packets and filtering routes.

Configuring Split Horizon

Split horizon can be used to avoid loop when multiple devices running distance-vectortype routing protocols connect to a network in which IP packets are broadcasted. Split horizon can prevent devices from advertising certain route information through an interface from which the devices learn such information. This optimizes route information exchange among multiple devices.

However, split horizon may cause the failure of some device to learn all the route information in a non-broadcast multi-access network (frame relay or X.25). In this case, you may need to disable split horizon. If an interface is configured with the secondary IP address, you need to pay attention to split horizon.

If the **poisoned-reverse** parameter is configured, split horizon with poisoned reverse is enabled. The device will advertise the route information from the interface where it learns the information, and configure the metric value of the route information as unreachable.

Use the following commands to enable or disable split horizon in interface configuration mode.

Command	Function
Ruijie(config-if)# **no ip split-horizon**	Disables split horizon.
Ruijie(config-if)# **ip split-horizon**	Enables split horizon.

Use the following commands to enable or disable split horizon with poisoned reverse in interface configuration mode.

Command	Function
Ruijie(config-if)# **no ip rip split-horizon poisoned-reverse**	Disable split horizon with poisoned reverse.
Ruijie(config-if)# **ip rip split-horizon poisoned-reverse**	Enable split horizon with poisoned reverse.

By default, all interfaces are configured as enabling split horizon without poisoned reverse.

Defining the RIP Version

Ruijie products support RIP version 1 and version 2, where RIPv2 supports authentication, key management, route convergence, CIDR, and VLSMs. For the information about key management and VLSMs, see the "*IP Routing Protocol Independent Feature Configuration*" chapter.

By default, Ruijie products can receive RIPv1 and RIPv2 packets, but they can send only RIPv1 packets. You can configure them to receive and send only RIPv1 or RIPv2 packets.

Use the following command to enable software to receive and send only the packets of a specific version in routing process configuration mode.

Command	Function
Ruijie(config-router)# **version** {**1** | **2**}	Defines the RIP version.

The above command allows the software to receive or send only the packets of a specific version by default. If necessary, you can modify the default setting of each interface.

Use the following commands to enable an interface to send only the packets of a specific version in interface configuration mode.

Command	Function
Ruijie(config-if)# **ip rip send version 1**	Specifies to send only RIPv1 packets.
Ruijie(config-if)# **ip rip send version 2**	Specifies to send only RIPv2 packets.
Ruijie(config-if)# **ip rip send version 1 2**	Specifies to send only RIPv1 and RIPv2 packets.

Use the following commands to configure an interface to receive only the packets of a specific version in interface configuration mode.

Command	Function
Ruijie(config-if)# **ip rip receive version 1**	Specifies to receive only RIPv1 packets.
Ruijie(config-if)# **ip rip receive version 2**	Specifies to receive only RIPv2 packets.
Ruijie(config-if)# **ip rip receive version 1 2**	Specifies to receive only RIPv1 and RIPv2 packets.

Configuring Route Convergence

Automatic RIP route convergence means that the routes of subnets are automatically converged into the routes of a classful network when they pass through the border of the classful network. By default, RIPv2 will automatically perform route convergence, while the RIPv1 does not support this function.

The automatic route convergence function of the RIPv2 improves the scalability and effectiveness of the network. If there are any converged routes, the sub-routes contained in them cannot be seen in the routing table. This greatly reduces the size of the routing table.

It is more efficient to advertise converged routes than the separated routes. Factors are as follows:

- Converged routes will be handled first when you search the RIP database.
- Any sub-routes will be ignored when you search the RIP database, and thus reducing the handling time.

Sometimes, you want to learn the specific sub-net routes rather than the converged network routes. In this case, you need to disable the automatic route convergence function.

Use the following commands to configure automatic route convergence in RIP routing progress mode.

Command	Function
Ruijie(config-router)# **no auto-summary**	Disables automatic route convergence.
Ruijie(config-router)# **auto-summary**	Enables automatic route convergence.

Use the following commands to configure interface-level convergence in interface mode. Then, configure route convergence within the specified classified subnet range on an interface.

Command	Function

Command	Function
Ruijie(config-if)# **ip summary-address** *rip ip-address ip-network-mask*	Enables route convergence on the interface.
Ruijie(config-if)# **no ip summary-address** *rip ip-address ip-network-mask*	Disables route convergence on the interface.

Configuring RIP Authentication

RIPv1 does not support authentication. If a device is configured with the RIPv2, you can configure authentication on an appropriate interface.

RIPv2 for Ruijie products supports two RIP authentication modes: plain-text authentication and MD5 authentication. The default authentication mode is plain-text authentication.

In plain-text authentication mode, you can run the **ip rip authentication text-password** command to configure the plain-text authentication password or obtain the plain-text authentication password through an associated key chain. The latter takes precedence over the former.

In MD5 authenticaiton mode, you must implement MD5 authentication through an associated key chain.

For plain-text authentication, no authentication occurs if no plain-text authentication password or associated key chain is configured. Similarly, for MD5 authentication, no authentication occurs if no associated key chain is configured.

If a key chain is specified in interface configuration mode, you need to use the **key chain** command in global configuration mode to define the key chain. Otherwise, authentication of RIP data packets may fail.

Use the following commands to configure RIP authentication in interface configuration mode.

Command	Function
Ruijie(config-if)# **ip rip authentication mode** {**text** | **md5**}	Uses the key chain, enables RIP authentication, and configures RIP authentication through the interface. text: indicates plain-text authentication. md5: indicates MD5 authentication.
Ruijie(config-if)# **ip rip authentication text-password** [**0**|**7**] *password-string*	Configures the plain-text authentication password in the length of 1 – 16 bytes. 0 Displays a key in plain text manner. 7 Displays a key in encrypted manner.
Ruijie(config-if)# **ip rip authentication key-chain** *key-chain-name*	Configures authentication using a key chain.

Configuring RIP Clock Adjustment

The RIP provides the clock adjustment function, which allows you to adjust a clock based on network conditions so that the RIP can run in a better way. You can adjust the following clocks:

Route update time: It defines the period in seconds for a device to send route update packets;

Route expiry time: It defines the time in seconds after which the routes in the routing table will become invalid if not updated;

Route clearing timer: It defines the time in seconds after which the routes in the routing table will be cleared;

By adjusting above clocks, the convergence and fault recovery of the routing protocol may be accelerated. Use the following command to adjust an RIP clock in RIP routing process configuration mode.

Command	Function
Ruijie(config-router)# **timers basic** *update invalid flush*	Adjusts the RIP clock.

By default, the update time is 30 seconds, the expiry time is 180 seconds, and the clearing time is 120 seconds.

Note For devices connected on the same network, the values of the RIP clocks must be the same.

Configuring Verification of the Source IP Address of an RIP Route

By default, the RIP will verify the source IP address of a received route update packet. The RIP will discard the packet if the source IP address is invalid. Judging whether the source IP address is valid, that is, judging whether the source IP address is in the same network as the IP address of the interface. No validation authentication will be performed on the interface of no numbered IP address.

Use the following commands to configure verification of route source IP address in RIP routing process configuration mode.

Command	Function
Ruijie(config-router)# **no validate-update-source**	Disables the source IP address validation.
Ruijie(config-router)# **validate-update-source**	Enables the source IP address validation.

Configuring Control of the RIP Interface Status

In some case, it is necessary to configure the RIP flexibly. If you only need to enable a device to learn RIP routes rather than advertising RIP routes, you can configure a passive interface. Or, if you need to configure the status of a certain interface individually, you can use a command to control the sending or receiving of the RIP packets on a specific interface.

Use the following commands to configure an interface as the passive interface in RIP routing process configuration mode.

Command	Function
Ruijie(config-router)# **passive-interface** {**default** | *interface-type interface-num*}	Configures a passive interface.
Ruijie(config-router)#**no passive-interface** {**default** | *interface-type interface-num*}	Cancels the configuration.

Caution A passive interface responds the non-RIP requests (such as the route diagnosis program) rather than the RIP requests because these request programs hope to learn about the routes of all devices.

Use the following commands to disable or allow an interface to receive RIP packets in interface configuration mode.

Command	Function

Command	Function
Ruijie(config-if)# **no ip rip receive enable**	Forbids the interface to receive the RIP packets.
Ruijie(config-if)# **ip rip receive enable**	Allows the interface to receive the RIP packets.

Use the following commands to disable or allow an interface to send RIP packets in interface configuration mode.

Command	Function
Ruijie(config-if)# **no ip rip send enable**	Forbids the interface to send the RIP packets.
Ruijie(config-if)# **ip rip send enable**	Allows the interface to send the RIP packets.

Configuring Default Route Advertisement through an Interface

Use the following command to generate a default route (0.0.0.0/0) in the update packet through a specified interface in interface configuration mode.

Command	Function
Ruijie(config-if)# **ip rip default-information originate** [**metric** *metric-value*]	Advertises the default route and other routes.
Ruijie(config-if)# **no ip rip default-information**	Cancels default route advertising through the interface.

In interface configuration mode, use the following commands to generate a default route (0.0.0.0/0) in the update route through a specified interface, and advertise only this default route instead of other RIP routes through this interface.

Command	Function
Ruijie(config-if)# **ip rip default-information only** [**metric** *metric-value*]	Advertises the default route only.
Ruijie(config-if)# **no ip rip default-information**	Cancels default route advertising through the interface.

If both the **ip rip default-information** command on the interface and the **default-information originate** command in the RIP progress are configured, only the default route configured on the interface is advertised.

Configuring Supernet Route Advertisement through the RIP Interface

A supernet route (for example, 80.0.0.0/6) is defined when the mask length is less than its natural mask length. According to IP address classification, 80.0.0.0 belongs to class-A network and its natural mask legnth is 8. Therefore, 80.0.0.0/6 is a supernet route.

When an RIPv1-enabled device monitors RIPv2 route response packets, it will learn incorrect routes because RIPv1 ignores the subnet masks of the routes in the packets if information about the supernet routes is received. In this case, an RIPv2-enabled device needs to disable advertising super network route on its interface.

Use the following command to configure whether to advertise the supernet route through an interfacein interface configuration mode.

Command	Function
Ruijie(config-if)#**no ip rip send supernet-routes**	Disables advertising the supernet route through the interface.
Ruijie(config-if)# **ip rip send supernet-routes**	Enables advertising the supernetwork route through the interface.

Note

1. When only RIPv1 packets rather than RIPv2 packets are received through the interface, no supernet route is received.
2. Supernet routes can be received when RIPv2 packets are allowed to be received through the interface.
3. No supernet route is sent when RIPv1 packets are sent through the interface.
4. Supernet routes are permitted to be sent by default when RIPv2 packets are sent through the interface.
5. The **no rip rip send supernet-routes** command prohibits sending supernet routes.

The **auto-summary** command takes no effect for supernet routes, that is, supernet routes are not converged.

The **ip rip summary** command does not support configuration of supernet routes.

Configuring RIP VRF

The RIP supports VRFs. Multiple RIP instances can be created to manage the corresponding VRFs in the RIP process. By default, there is only one RIP instance in the RIP process, which is used to manage the global routing table. After a VRF is created, you can manage the routing table of the VRF by creating a new RIP instance.

Run the **address-family** command to enable a router device to enter the address family configuration mode (with the prompt (config-router-af)#). When you specify the VRF associated with the sub mode at the first time, the RIP will create a RIP instance corresponding to the VRF. Under this mode, you can configure the RIP instance of the VRF in the same way as that in global route configuration mode.

To exit the address family configuration sub mode and return to the route configuration mode, run the **exit-address-family** command.

Use the following commands to configure a RIP instance managng the VRF in RIP routing process configuration mode.

Command	Function
Ruijie(config-router)# **address-family ipv4 vrf** *vrf-name*	Creates the RIP instance managing the VRF.
Ruijie(config-router)#**no address-family ipv4 vrf** *vrf-name*	Removes the RIP instance managing the VRF.

Configuring RIP BFD

For details on RIP BFD configuration, see *BFD Configuration Guide*.

Configuring TRIP

Triggered RIP (TRIP) is a RIP extension on a wide area network(WAN), and is mainly used on the on-demand link.

When TRIP is enabled, RIP protocol will no longer periodically send route updates but only send route updates to WAN interfaces in the following cases:

- When route update request packets are received.
- When RIP routing information has changed.
- When interface state changes.
- When routers start.

Since the periodic RIP update is canceled, an acknowledgement and retransmission mechanism is required to guarantee successful update packet transmission and receiving on the WAN. RIP uses three new types of packets which are identified by the value of the command field in the RIP header:

- Update request (Type-9): requests the peer to send the routing information needed.
- Update response (Type-10): contains the route updates requested by the peer.
- Update Acknowledge (Type-11): acknowledges the received update responses, indicating that the route updates sent by peer have been received.

Caution

1. This function can be used in the following cases: (1) The interface is connected to only one neighbor; (2) The interface is connected to multiple neighbors using unicast communication mode. You are advised to enable this feature on PPP, frame relay, X.25, and similar link layer protocols.
2. You are advised to enable split horizon with poisoned reverse on TRIP-enabled interface. Otherwise, there may be residual invalid routing information.
3. It shall be guaranteed that the feature is enabled on all routers on the same link. Otherwise, the function may fail and routing information cannot be exchanged properly.
4. This function cannot be used together with BFD for RIP;
5. When this function is enabled, make sure that the RIP configurations on both ends of the link are identical, such as RIP authentication and version of RIP protocol supported by the interface and etc.
6. With this function enabled on the interface, the valid-update-source will be performed for the packets of this interface no matter whether the valid-update-source function is enabled.

Use the following commands to enable or disable this function in interface configuration mode.

Command	Function
Ruijie(config-if)# **ip rip triggered**	Enables Triggered RIP.
Ruijie(config-if)# **no ip rip triggered**	Disables Triggered RIP.

Configuring RIP GR

RIP graceful restart (GR) guarantees non-stop data forwarding during the process of protocol restart. When RIP GR is enabled on the router, the forwarding table will be maintained during the process of RIP restart, and request packets will be sent to neighbors to re-learn routes in order to complete route re-convergence within the period of graceful restart. Upon expiration of the GR period, GR will exit and forwarding table entries will be updated and advertised to neighbors.

The GR period is the maximum duration from RIP GR execution to RIP GR completion. During this period, the forwarding table will be maintained and RIP route recovery will be implemented in order to restore RIP to the state before GR. Upon expiration of grace period, RIP will exit from the GR state and perform common RIP operations.

graceful-restart grace-period allows users to explicitly change the restart period. Please note that GR must be completed within the RIP expiration time and one RIP route update cycle is completed. If this value is not properly configured, non-stop data forwarding cannot be guaranteed during the GR process. For example, if the GR period is longer than the expiration time of neighbor routers and GR is not completed within such expiration time, the neighbor's routes will not be sent upon expiration of the expiration time, thus causing interruption of data forwarding. Therefore, unless otherwise specified, it is not allowed to adjust the GR period. If the GR period is adjusted, please refer to the configuration of the **timers basic** command and make sure the GR period is longer than the update time and smaller than the expiration time.

Use the following commands to enable or disable this function in RIP routing process configuration mode.

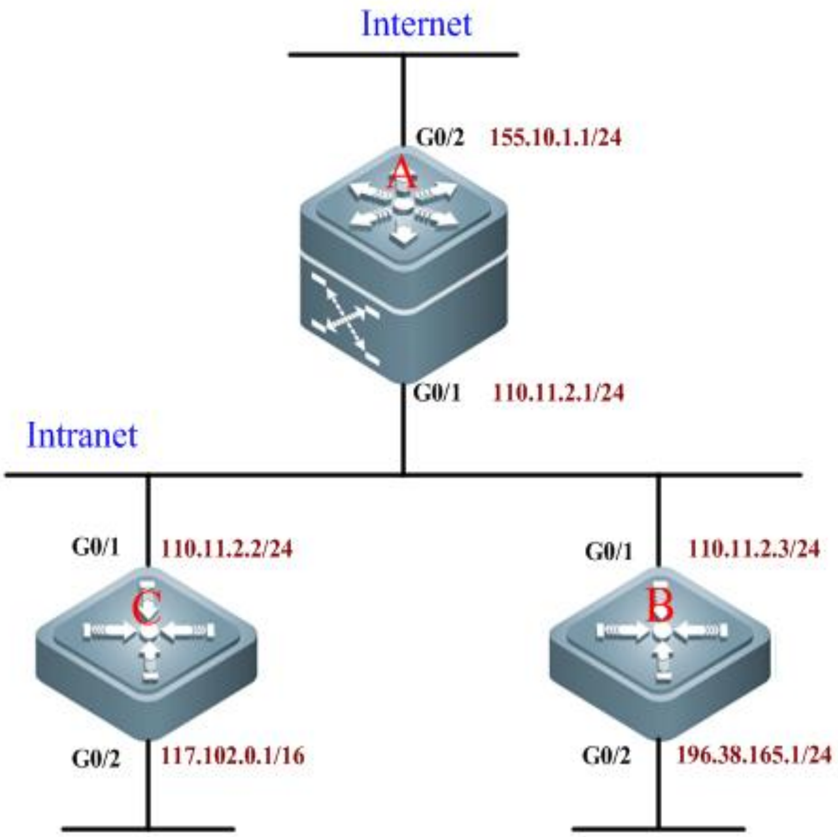
Command	Function
Ruijie(config-router)# **graceful-restart** [**grace-period** *grace-period*]	Enables RIP GR.
Ruijie(config-router)# **no graceful-restart** [**grace-period**]	Disables RIP GR.

RIP Configuration Examples

Configuring RIP Routes and Defining RIP Versions

Networking Topology

Figure 1 Configuring RIP routes and defining RIP versions



Networking Requirements

A small-sized company runs on a small office network, and requires network layer intercommunication between any two nodes. Networking requirements are as follows:

- Devices shall be able to adapt to the changes in the network topology, in order to reduce the workload of manual maintenance;
- Route updates can carry subnet masks;
- Device A only receives the routing information from external networks, but will not advertise routing information of internal network.
- RIP information can be exchanged between devices A, B, and C, so that internal hosts can access Internet.

Configuration Tips

- According to user's requirements and network environment, the RIPv2 routing protocol is selected to achieve user network intercommunication;
- To allow device A to receive routing information sent from external network without advertising the routing information of internal network, the G0/2 port of device A shall be configured as a passive interface.

Configuration Steps

# Configure device A

! Configure the IP address of the corresponding port on device A.

```
Ruijie>enable
Ruijie#configure terminal
Ruijie(config)#interface gigabitEthernet 0/1
Ruijie(config-if)#ip address 110.11.2.1 255.255.255.0
Ruijie(config-if)#exit
Ruijie(config)#interface gigabitEthernet 0/2
Ruijie(config-if)#ip address 155.10.1.1 255.255.255.0
```

! Create the RIP routing progress.

```
Ruijie(config)#router rip
```

! Configure RIP version as version 2.

```
Ruijie(config-router)#version 2
```

! Configure G0/2 as a passive interface.

```
Ruijie(config-router)#passive-interface gigabitEthernet 0/2
```

! Disable automatic route convergence.

```
Ruijie(config-router)#no auto-summary
```

! Specify the associated network.

```
Ruijie(config-router)#network 110.11.2.0 255.255.255.0
Ruijie(config-router)#network 155.10.1.0
```

# Configure device B

! Configure the IP address of the corresponding port on device B.

```
Ruijie>enable
Ruijie#configure terminal
Ruijie(config)#interface gigabitEthernet 0/1
Ruijie(config-if)#ip address 110.11.2.2 255.255.255.0
Ruijie(config-if)#exit
Ruijie(config)#interface gigabitEthernet 0/2
Ruijie(config-if)#ip address 196.38.165.1 255.255.255.0
Ruijie(config-if)#exit
```

! Create RIP routing progress.

```
Ruijie(config)#router rip
```

! Configure the RIP version as version 2.

```
Ruijie(config-router)#version 2
```

! Disable automatic route convergence.

```
Ruijie(config-router)#no auto-summary
```

! Specify the associated network.

```
Ruijie(config-router)#network 110.11.2.0
Ruijie(config-router)#network 196.38.165.0
```

# Configure device C

! Configure the IP address of the corresponding port on device C.

```
Ruijie>enable
Ruijie#configure terminal
Ruijie(config)#interface gigabitEthernet 0/1
Ruijie(config-if)#ip address 110.11.2.3 255.255.255.0
Ruijie(config-if)#exit
Ruijie(config)#interface gigabitEthernet 0/2
Ruijie(config-if)#ip address 117.102.0.1 255.255.0.0
Ruijie(config-if)#exit
```

! Create RIP routing progress.

```
Ruijie(config)#router rip
```

! Configure RIP version as version 2.

```
Ruijie(config-router)#version 2
```

! Disable automatic route convergence.

```
Ruijie(config-router)#no auto-summary
```

! Specify the associated network.

```
Ruijie(config-router)#network 110.11.2.0
Ruijie(config-router)#network 117.102.0.0
```

Verification

View the routing table of each device;

View the routing table on A, as shown below (the bold figures are the routing information learned through RIP):

```
Ruijie#show ip route
Codes:  C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
C    110.11.2.0/24 is directly connected, GigabitEthernet 0/1
C    110.11.2.1/32 is local host.
R 117.102.0.0/16 [120/1] via 110.11.2.2, 00:00:47, GigabitEthernet 0/1
C    155.10.1.0/24 is directly connected, GigabitEthernet 0/2
C    155.10.1.1/32 is local host.
C    192.168.217.0/24 is directly connected, VLAN 1
C    192.168.217.233/32 is local host.
R  196.38.165.0/24 [120/1] via 110.11.2.3, 00:19:18, GigabitEthernet 0/1
```

View the routing table on B, as shown below (the bold figures are the routing information learned through RIP):

```
Ruijie#show ip route

Codes:  C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
C    110.11.2.0/24 is directly connected, GigabitEthernet 0/1
C    110.11.2.2/32 is local host.
R 155.10.1.0/24 [120/1] via 110.11.2.1, 00:15:21, GigabitEthernet 0/1
C    196.38.165.0/24 is directly connected, GigabitEthernet 0/2
C    196.38.165.1/32 is local host.
R 117.102.0.0/16 [120/1] via 110.11.2.2, 00:00:47, GigabitEthernet 0/1
```

View the routing table on C, as shown below (the bold figures are the routing information learned through RIP):

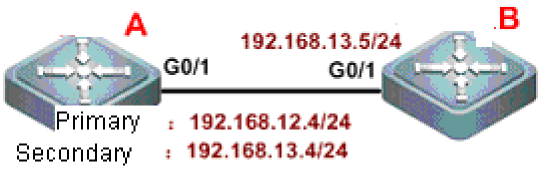
```
Ruijie#show ip route

Codes:  C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
C    110.11.2.0/24 is directly connected, GigabitEthernet 0/1
C    110.11.2.3/32 is local host.
```

```
C    117.102.0.0/16 is directly connected, GigabitEthernet 0/2
C    117.102.0.1/32 is local host.
R  155.10.1.0/24 [120/1] via 110.11.2.1, 00:20:55, GigabitEthernet 0/1
R  196.38.165.0/24 [120/1] via 110.11.2.3, 00:19:18, GigabitEthernet 0/1
```

RIP Split Horizon

Networking Topology

Figure 2 Topology for RIP split horizon



Networking Requirements

There are two devices on the network. Device A is configured with a secondary IP address.

The following requirements shall be met:

- The RIP routing protocol runs on both devices;
- Device B can learn the routes of network segment 192.168.12.0/24.

Configuration Tips

To meet the above requirements, the following configurations are required:

- RIPv2 routing protocol is run on both devices;
- Split horizon shall be disabled on device A (by default, split horizon is enabled on all interfaces), or else device A won't advertise network segment 192.168.12.0 to device B.

Configuration Steps

# Configure device A

! Configure Ethernet ports.

```
Ruijie(config)#interface gigabitEthernet 0/1
Ruijie(config-if-GigabitEthernet 0/1)#ip address 192.168.12.4 255.255.255.0
Ruijie(config-if-GigabitEthernet 0/1)#ip address 192.168.13.4 255.255.255.0 secondary
```

! Disable split horizon.

```
Ruijie(config-if-GigabitEthernet 0/1)#no ip rip split-horizon
```

! Configure the RIP routing protocol.

```
Ruijie(config)#route rip
Ruijie(config-router)#version 2
Ruijie(config-router)#network 192.168.12.0
Ruijie(config-router)#network 192.168.13.0
```

! Disable automatic route convergence.

```
Ruijie(config-router)#no auto-summary
```

# Configure device B

! Configure Ethernet ports.

```
Ruijie(config)#interface gigabitEthernet 0/1
Ruijie(config-if-GigabitEthernet 0/1)#ip address 192.168.13.5 255.255.255.0
```

! Configure the RIP routing protocol.

```
Ruijie(config)#route rip
Ruijie(config-router)#version 2
Ruijie(config-router)#network 192.168.13.0
```

Verification

View the routing table on device B before and after disabling split horizon.

Before split horizon is disabled, view the routing table on device B, as shown below:

```
Ruijie#show ip route

Codes:  C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default
Gateway of last resort is no set
C    192.168.13.0/24 is directly connected, GigabitEthernet 0/1
C    192.168.13.5/32 is local host.
```

After split horizon is disabled, view the routing table on device B, as shown below (the bold figures are the routing information learned through RIP):

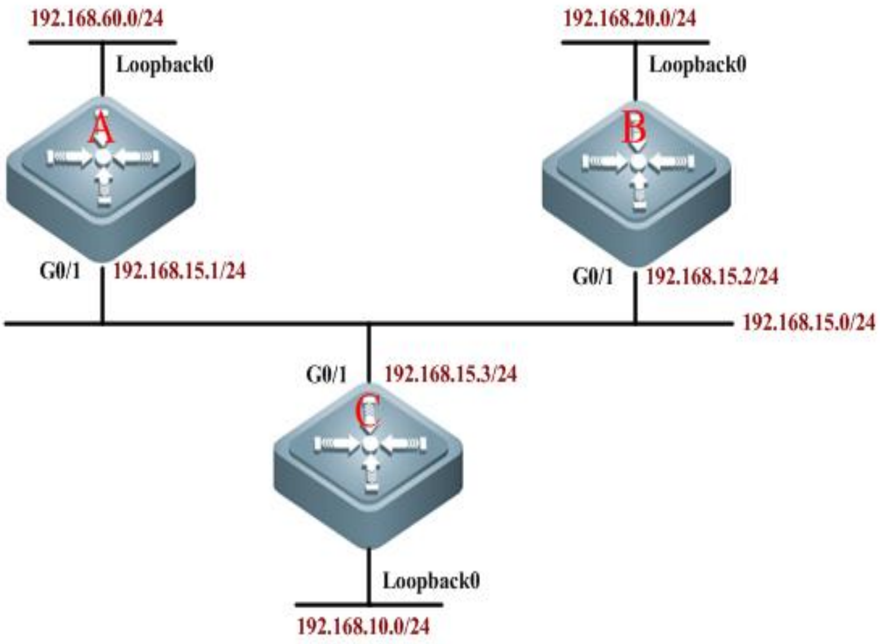
```
Ruijie#show ip route

Codes:  C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
R    192.168.12.0/24 [120/1] via 192.168.13.4, 00:00:10, GigabitEthernet 0/1
```

```
C    192.168.13.0/24 is directly connected, GigabitEthernet 0/1
C    192.168.13.5/32 is local host.
```

RIP Unicast Update

Networking Topology

Figure 3 Topology for RIP unicast update



Networking Requirements

As shown below, three devices are connected to the LAN and run the RIP routing protocol.

- Device A can learn the routes advertised by devices B and C;
- Device C can learn the routes advertised by devices A and B;
- Device B cannot learn the routes advertised by device C.

Configuration Tips

To meet the above configuration requirements, RIP unicast packets must be configured on device C. Add the command of **neighbor** during the RIP configuration of device C, so that the RIP protocol can send advertisements to the interface of device A in unicast mode. Configure the **passive-interface** command on G0/1 of device C to avoid broadcast update on this link.

Configuration Steps

# Configure device A

! Configure the IP address of corresponding interface.

```
Ruijie>enable
Ruijie#configure terminal
```

```
Ruijie(config)#interface gigabitEthernet 0/1
Ruijie(config-if)#ip address 192.168.15.1 255.255.255.0
Ruijie(config-if)#exit
Ruijie(config)#interface Loopback 0
Ruijie(config-if)#ip address 192.168.60.1 255.255.255.0
Ruijie(config-if)#exit
```

! Create RIP routing progress.

```
Ruijie(config)#router rip
```

! Specify the associated network.

```
Ruijie(config-router)#network 192.168.60.0
Ruijie(config-router)#network 192.168.15.0
```

# Configure device B

! Configure the IP address of corresponding interface.

```
Ruijie>enable
Ruijie#configure terminal
Ruijie(config)#interface gigabitEthernet 0/1
Ruijie(config-if)#ip address 192.168.15.2 255.255.255.0
Ruijie(config-if)#exit
Ruijie(config)#interface Loopback 0
Ruijie(config-if)#ip address 192.168.20.1 255.255.255.0
Ruijie(config-if)#exit
```

! Create RIP routing progress.

```
Ruijie(config)#router rip
```

! Specify the associated network.

```
Ruijie(config-router)#network 192.168.20.0
Ruijie(config-router)#network 192.168.15.0
```

# Configure device C

! Configure the IP address of corresponding interface.

```
Ruijie>enable
Ruijie#configure terminal
Ruijie(config)#interface gigabitEthernet 0/1
Ruijie(config-if)#ip address 192.168.15.3 255.255.255.0
Ruijie(config-if)#exit
Ruijie(config)#interface Loopback 0
Ruijie(config-if)#ip address 192.168.10.1 255.255.255.0
Ruijie(config-if)#exit
```

! Create RIP routing progress.

```
Ruijie(config)#router rip
```

! Specify the associated network.

```
Ruijie(config-router)#network 192.168.15.0
Ruijie(config-router)#network 192.168.10.0
```

! Configure G0/1 as a passive interface.

```
Ruijie(config-router)#passive-interface gigabitEthernet 0/1
```

! Enable unicast update.

```
Ruijie(config-router)#neighbor 192.168.15.1
```

Verification

View the routing table of each device (mainly the routing information on devices C and B):

View the routing table on device B, as shown in the following figure:

```
Ruijie#show ip route

Codes:  C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
C    192.168.20.0/24 is directly connected, Loopback 0
C    192.168.20.1/32 is local host.
C    192.168.15.0/24 is directly connected, GigabitEthernet 0/1
C    192.168.15.2/32 is local host.
R 192.168.60.0/24 [120/1] via 192.168.15.1, 00:15:21, GigabitEthernet 0/1
```

View the routing table on device C, as shown below (the bold figures are the routing information learned through RIP):

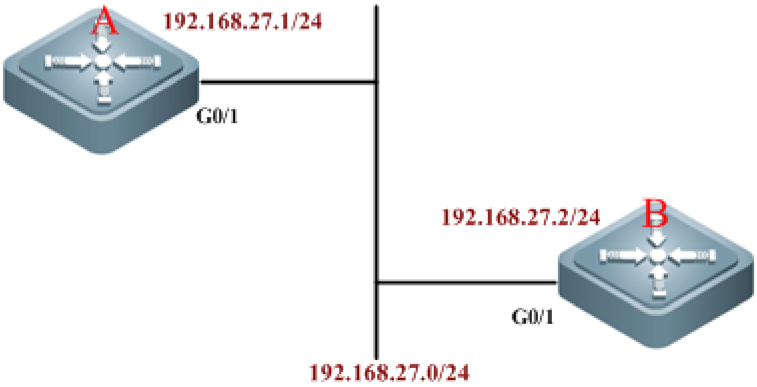
```
Ruijie#show ip route

Codes:  C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
C    192.168.10.0 is directly connected, Loopback 0
C    192.168.10.1/32 is local host.
R 192.168.60.0/24 [120/1] via 192.168.15.1, 00:15:21, GigabitEthernet 0/1
C    192.168.15.0/24 is directly connected, GigabitEthernet 0/1
```

```
C    192.168.15.3/32 is local host.
R 192.168.20.0 [120/1] via 192.168.15.2, 00:00:47, GigabitEthernet 0/1
```

RIP Authentication

Networking Topology

Figure 4 Topology for RIP authentication



Networking Requirements

Interconnected through Ethernet, two devices run the RIP routing protocol and use MD5 authentication. The requirements are as follows:

- The authentication key for device A to send RIP packets is "Hello", and device A can receive RIP packets with authentication keys being "Hello" and "World";
- The authentication key for device B to send RIP packets is "World", and device B can receive RIP packets with authentication keys being "Hello" and "World";
- The first key is used from 4:30pm October 1st, 2010 for 12 hours (43200s)
- The second key becomes permanently valid from 4:00 am October 2, 2010.

Configuration Tips

Authentication is not supported in RIPv1. If the RIPv2 routing protocol is configured on the device, authentication can then be configured on the corresponding interface.

The key string specifies the key set that can be used by this interface. If the key string is not configured and even if the interface uses the key chain, no authentication occurs. Therefore, before configuring authentication, the key chain and the associated key string must be configured first.

RGOS supports two RIP authentication modes: plain text and MD5, while plain text is the default authentication mode.

- The authentication key for sending RIP packets must be configured with the first key on keychain;
- When configuring the authentication key that can be received, configure any key on the keychain.

Configuration Steps

# Configure device A:

! Configure the IP address of Ethernet interface.

```
Ruijie>enable
Ruijie#configure terminal
Ruijie(config)#interface gigabitEthernet 0/1
Ruijie(config-if)#ip address 192.168.27.1 255.255.255.0
Ruijie(config-if)#exit
```

! Configure the key chain named "ripchain".

```
Ruijie(config)#key chain ripchain
```

! Configure the first key of "Key 1", which contains the key-string of "Hello", and configure the corresponding period needed.

```
Ruijie(config-keychain)#key 1
Ruijie(config-keychain-key)#key-string Hello
Ruijie(config-keychain-key)#accept-lifetime 16:30:00 Oct 1 2010 duration 43200
Ruijie(config-keychain-key)#send-lifetime 16:30:00 Oct 1 2010 duration 43200
Ruijie(config-keychain-key)#exit
```

! Configure the second key of "Key 2", which contains the key-string of "World", and configure the corresponding period needed.

```
Ruijie(config-keychain)#key 2
Ruijie(config-keychain-key)#key-string World
Ruijie(config-keychain-key)#accept-lifetime 04:00:00 Oct 2 2010 infinite    //Beginning time that the key is valid to be received
Ruijie(config-keychain-key)#send-lifetime 04:00:00 Oct 2 2010 infinite        //Beginning time that the key is valid to be sent
Ruijie(config-keychain-key)#end
```

! Configure G0/1 to use the MD5 authentication key to authenticate the update messages sent from device B.

```
Ruijie#configure terminal
Ruijie(config)#interface gigabitEthernet 0/1
Ruijie(config-if)#ip rip authentication key-chain ripchain
Ruijie(config-if)#ip rip authentication mode md5
Ruijie(config-if)#exit
```

! Configure the RIP routing protocol.

```
Ruijie(config)#router rip
Ruijie(config-router)#version 2
Ruijie(config-router)#network 192.168.27.0
```

#Configure device B:

! Configure the IP address of Ethernet interface.

```
Ruijie>enable
Ruijie#configure terminal
Ruijie(config)#interface gigabitEthernet 0/1
Ruijie(config-if)#ip address 192.168.27.2 255.255.255.0
Ruijie(config-if)#exit
```

! Configure the key chain.

```
Ruijie(config)#key chain ripchain //The name of key chain is only valid on the local device.
You can also use other names.
```

! Configure the first key of "Key 1", which contains the key-string of "Hello", and configure the corresponding period needed.

```
Ruijie(config-keychain)#key 1
Ruijie(config-keychain-key)#key-string Hello
Ruijie(config-keychain-key)#accept-lifetime 16:30:00 Oct 1 2010 duration 43200
Ruijie(config-keychain-key)#send-lifetime 16:30:00 Oct 1 2010 duration 43200
Ruijie(config-keychain-key)#exit
```

! Configure the second key of "Key 2", which contains the key-string of "World", and configure the corresponding period needed.

```
Ruijie(config-keychain)#key 2
Ruijie(config-keychain-key)#key-string World
Ruijie(config-keychain-key)#accept-lifetime 04:00:00 Oct 2 010 infinite
Ruijie(config-keychain-key)#send-lifetime 04:00:00 Oct 2 2010 infinite
Ruijie(config-keychain-key)#end
```

! Configure G0/1 to use the MD5 authentication key to authenticate the update messages sent from device A.

```
Ruijie#configure terminal
Ruijie(config)#interface gigabitEthernet 0/1
Ruijie(config-if)#ip rip authentication key-chain ripchain
Ruijie(config-if)#ip rip authentication mode md5
Ruijie(config-if)#exit
```

! Configure the RIP routing protocol.

```
Ruijie(config)#router rip
Ruijie(config-router)#version 2
Ruijie(config-router)#network 192.168.27.0
```

Verification

Run the show run command to verify the correctness of configurations (taking device A as the example):

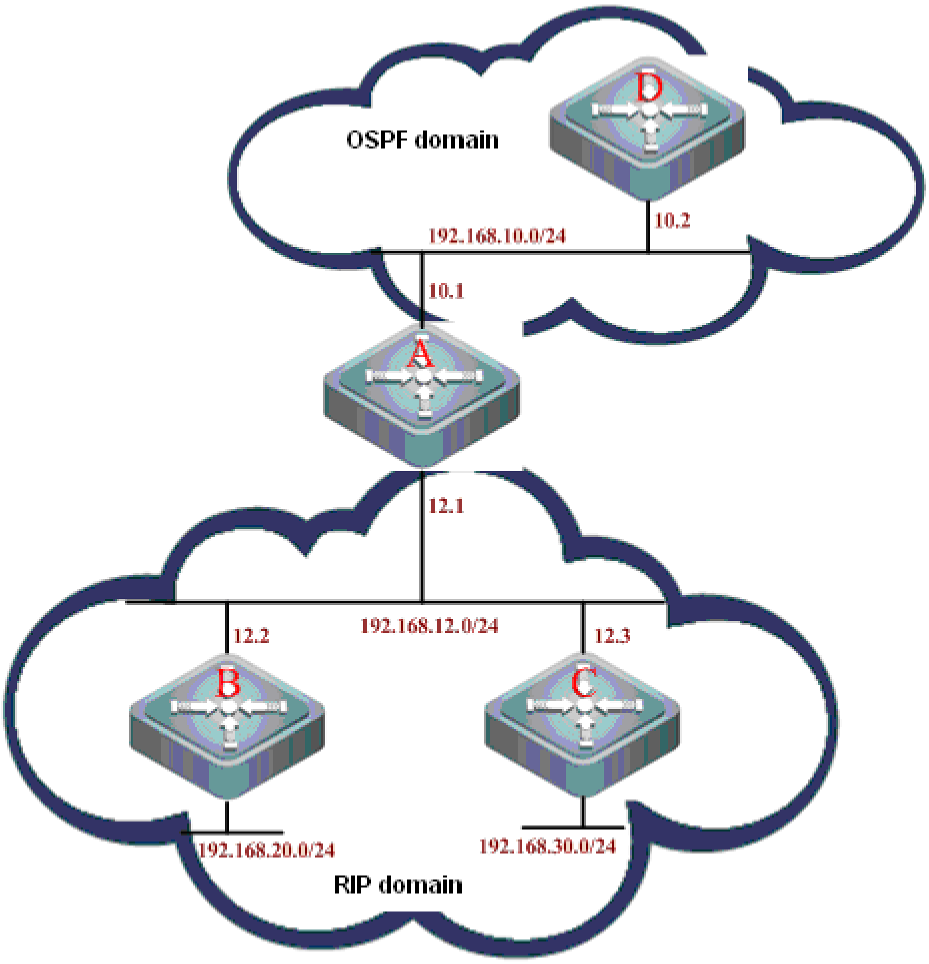
```
Ruijie#show run

Building configuration...
Current configuration : 1561 bytes

!
vlan 1
!
```

```
!
key chain ripchain
 key 1
  key-string Hello
    accept-lifetime 16:30:00 Oct 01 2010 duration 43200
  send-lifetime 16:30:00 Oct 01 2010 duration 43200
 key 2
  key-string World
    accept-lifetime 04:00:00 Oct 02 2010 infinite
    send-lifetime 04:00:00 Oct 02 2010 infinite
!
no service password-encryption
!
interface GigabitEthernet 0/1
     ip rip authentication mode md5
 ip rip authentication key-chain ripchain
 no ip proxy-arp
    ip address 192.168.27.1 255.255.255.0
!
interface GigabitEthernet 0/2
!
interface GigabitEthernet 0/3
!
interface GigabitEthernet 0/4
...
!
!
!
router rip
 version 2
 network 192.168.27.0
!
!
!
line con 0
line vty 0 4
 login
!
!
end
```

RIP Redistribution and Default Route

Networking Topology

Figure 5 Topology for RIP redistribution and default route



Networking Requirements

Devices A, B, and C are interconnected in the same network segment and run the RIP routing protocol. Devices A and D are interconnected in the same network segment and run the OSPF routing protocol. Configure these four devices to achieve the following goals:

- Device A can learn the OSPF routes advertised by device D;
- Device A can redistribute OSPF routes to RIP;
- Device A advertises the redistributed routes to devices B and C;
- Device C advertises the default routing to devices A and B.

Configuration Tips

- Configure to redistribute OSPF routes to RIP in the RIP process of device A;
- Configure to advertise the default routing on the corresponding interface of device C;

Configuration Steps

#Configure device A:

! Configure Ethernet ports.

```
Ruijie(config)#interface FastEthernet 0/1
Ruijie(config-if-FastEthernet 0/1)#ip address 192.168.12.1 255.255.255.0
Ruijie(config-if-FastEthernet 0/1)#exit
Ruijie(config)#interface FastEthernet0/2
Ruijie(config-if-FastEthernet 0/2)#ip address 192.168.10.1 255.255.255.0
```

# Configure the RIP routing protocol.

```
Ruijie(config)#router rip
Ruijie(config-router)#version 2
Ruijie(config-router)#network 192.168.12.0
Ruijie(config-router)#redistribute ospf 10 metric 3
```

//Redistribute the OSPF routing progress in the RIP progress, with metric value being 3

# Configure the OSPF routing protocol.

```
Ruijie(config)#router ospf 10
Ruijie(config-router)#network 192.168.10.0 0.0.0.255 area 0
```

#Configure device B:

! Configure Ethernet ports.

```
Ruijie(config)#interface FastEthernet 0/1
Ruijie(config-if-FastEthernet 0/1)#ip address 192.168.12.2 255.255.255.0
Ruijie(config-if-FastEthernet 0/1)#exit
```

! Configure loopback ports.

```
Ruijie(config)#interface Loopback 0
Ruijie(config-if-Loopback 0)#ip address 192.168.20.1 255.255.255.0
```

! Configure the RIP routing protocol.

```
Ruijie(config)#router rip
Ruijie(config-router)#version 2
Ruijie(config-router)#network 192.168.12.0
Ruijie(config-router)#network 192.168.20.0
```

# Configure device C:

! Configure Ethernet ports.

```
Ruijie(config)#interface FastEthernet 0/1
Ruijie(config-if-FastEthernet 0/1)#ip address 192.168.12.3 255.255.255.0
Ruijie(config-if-FastEthernet 0/1)#ip rip default-information originate metric 5
```

//Advertise default route, with metric value being 5

! Configure loopback ports.

```
Ruijie(config)#interface Loopback 0
Ruijie(config-if-Loopback 0)#ip address 192.168.30.1 255.255.255.0
```

# Configure the RIP routing protocol.

```
Ruijie(config)#router rip
Ruijie(config-router)#version 2
Ruijie(config-router)#network 192.168.12.0
Ruijie(config-router)#network 192.168.30.0
```

# Configure device D:

! Configure Ethernet ports.

```
Ruijie(config)#interface FastEthernet 0/1
Ruijie(config-if-FastEthernet 0/1)#ip address 192.168.10.2 255.255.255.0
```

! Configure the OSPF routing protocol.

```
Ruijie(config)#router ospf 10
Ruijie(config-router)#network 192.168.10.0 0.0.0.255 area 0
```

Verification

View the routing table of each device (mainly the routing information on devices A, B, and C):

View the routing table on device A, as shown below (the bold figures are the routing information learned through RIP):

```
Ruijie#show ip route

Codes:  C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
R*   0.0.0.0/0 [120/5] via 192.168.12.3, 00:00:23, FastEthernet 0/1
C    192.168.10.0/24 is directly connected, FastEthernet 0/2
C    192.168.10.1/32 is local host.
C    192.168.12.0/24 is directly connected, FastEthernet 0/1
C    192.168.12.1/32 is local host.
R    192.168.20.0/24 [120/1] via 192.168.12.2, 00:07:09, FastEthernet 0/1
R    192.168.30.0/24 [120/1] via 192.168.12.3, 00:00:23, FastEthernet 0/1
```

View the routing table on device B, as shown below (the bold figures are the routing information learned through RIP):

```
Ruijie#show ip route

Codes:  C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default
```

```

Gateway of last resort is no set
R    192.168.10.0/24 [120/3] via 192.168.12.1, 00:00:06, FastEthernet 0/1
C    192.168.12.0/24 is directly connected, FastEthernet 0/1
C    192.168.12.2/32 is local host.
C    192.168.20.0/24 is directly connected, Loopback 0
C    192.168.20.1/32 is local host.
R    192.168.30.0/24 [120/3] via 192.168.12.3, 00:00:06, FastEthernet 0/1
```

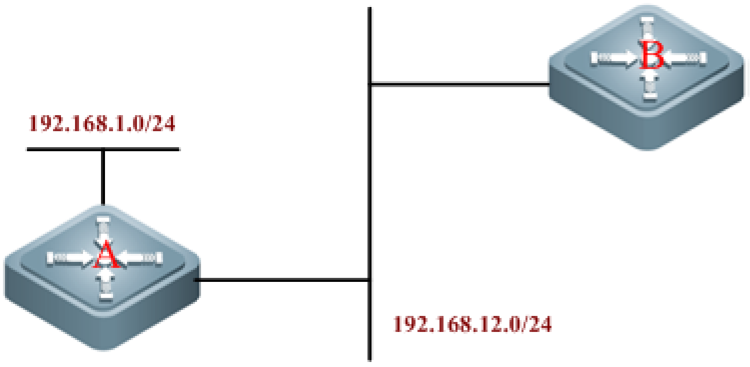
View the routing table on device C, as shown below (the bold figures are the routing information learned through RIP):

```
Ruijie#show ip route

Codes:  C - connected, S - static, R - RIP, B - BGP
        O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
R    192.168.10.0/24 [120/3] via 192.168.12.1, 00:01:49, FastEthernet 0/1
C    192.168.12.0/24 is directly connected, FastEthernet 0/1
C    192.168.12.3/32 is local host.
C    192.168.30.0/24 is directly connected, Loopback 0
C    192.168.30.1/32 is local host.
R    192.168.20.0/24 [120/3] via 192.168.12.2, 00:01:49, FastEthernet 0/1
```

RIP Supernet Route

Networking Topology

Figure6 Topology for the RIP supernet route



Networking Requirements

Two devices are interconnected through Ethernet. Device A runs RIPv2, and device B only supports the RIPv1 protocol and is unable to learn supernet routes.

Requirements:

- Configure supernet route 80.0.0.0/6 on device A, with next hop pointing to interface loopback 1 (192.168.1.0);
- Redistribute the aforementioned static route to RIP;
- Prohibit advertising supernet routes on device A.

Configuration Tips

Device B supports only the RIPv1 protocol. According to RFC 1058, such device is able to receive update packets of higher-version RIP, but such fields as subnet mask and next hop in the packets must be neglected. Therefore, route 80.0.0.0/6 received by device B will be treated as 80.0.0.0/8. To prevent device B from learning incorrect routes, device A must be configured to prohibit supernet route advertisement.

Configuration Steps

# Configure device A:

! Configure Ethernet ports.

```
Ruijie(config)#interface FastEthernet 0/1
Ruijie(config-if-FastEthernet 0/1)#ip address 192.168.12.1 255.255.255.0
Ruijie(config-if-FastEthernet 0/1)#no ip rip send supernet-routes
```

//Prohibit supernet route advertisement

! Configure loopback ports.

```
Ruijie(config)#interface loopback 1
Ruijie(config-if-Loopback 1)#ip address 192.168.1.1 255.255.255.0
```

! Configure static routes.

```
Ruijie(config)#ip route 80.0.0.0 252.0.0.0 loopback 1
```

! Configure the RIP routing protocol.

```
Ruijie(config)#router rip
Ruijie(config-router)#version 2
Ruijie(config-router)#network 192.168.12.0
Ruijie(config-router)#network 192.168.1.0
Ruijie(config-router)#redistribute static
```

//Redistribute static route

# Configure device B (supporting RIPv1 only):

! Configure Ethernet ports.

```
Ruijie(config)#interface FastEthernet 0/1
Ruijie(config-if-FastEthernet 0/1)#ip address 192.168.12.3 255.255.255.0
```

! Configure the RIP routing protocol.

```
Ruijie(config)#router rip
Ruijie(config-router)#network 192.168.12.0
```

Verification

View the routing table of each device;

View the routing table on device A, as shown below:

```
Ruijie#show ip route

Codes:  C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
S    80.0.0.0/6 is directly connected, Loopback 1
C    192.168.1.0/24 is directly connected, Loopback 1
C    192.168.1.1/32 is local host.
C    192.168.12.0/24 is directly connected, FastEthernet 0/1
C    192.168.12.1/32 is local host.
```

View the routing table on device B, as shown below (the bold figures are the routing information learned through RIP):

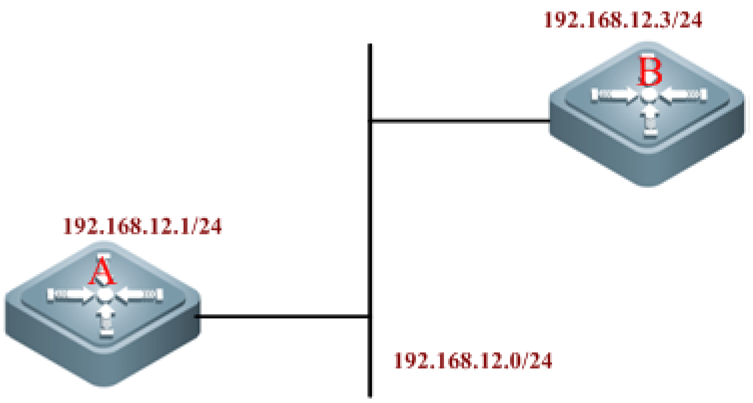
```
Ruijie#show ip route

Codes:  C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
R    80.0.0.0/6 [120/1] via 192.168.12.1, 00:00:46, GigabitEthernet 0/1
R    192.168.1.0/24 [120/1] via 192.168.12.1, 00:38:17, FastEthernet 0/1
C    192.168.12.0/24 is directly connected, FastEthernet 0/1
C    192.168.12.2/32 is local host.
```

RIP VRF Configuration Examples

Networking Requirements

Two routing devices are interconnected through Ethernet and run the RIP routing protocol. The connection layout and IP address distribution are shown in Figure 7.

Figure7 Example of RIP VRF configuration



Through RIP, routing information is exchanged between VRF "redvpn" of device A and VRF "bluevpn" of device B.

By enabling RIP GR on device A and setting the GR period to 90 seconds, non-stop data forwarding can be realized during hot standby switchover between main and slave management boards on device A. Meanwhile, since the GR period has been changed, timers basic shall be configured to a reasonable value.

Detailed Configurations

Configure device A:

# Create VRF.

```
ip vrf redvpn
```

# Bind the interface to VRF and configure the interface IP address.

```
interface fastEthernet 0/1
ip vrf forwarding redvpn
ip address 192.168.12.1 255.255.255.0
```

# Configure the RIP routing protocol and create a RIP instance.

```
router rip
address-family ipv4 vrf redvpn
network 192.168.12.0
graceful-restart grace-period 90
timers basic 45 270 180
exit-address-family
```

Configure device B:

# Create VRF.

```
ip vrf  bluevpn
```

# Bind the interface to VRF and configure the interface IP address.

```
interface fastEthernet 0/1
ip vrf forwarding bluevpn
ip address 192.168.12.3 255.255.255.0
```

# Configure the RIP routing protocol and create a RIP instance.

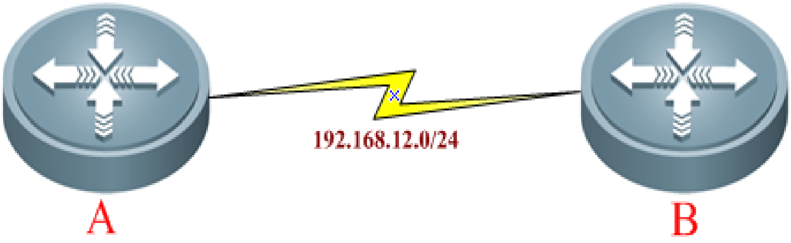
```
router rip
address-family ipv4 vrf bluevpn
network 192.168.12.0
timers basic 45 270 180
exit-address-family
```

TRIP Configuration Examples

Networking Requirements

Two routers are interconnected through the PPP link and run the RIP routing protocol. The connection layout and IP address distribution are shown in Figure 8.

Figure 8 Example of TRIP configuration



By configuring TRIP, routing information can be exchanged between devices A and B on the WAN link, and split horizon with poisoned reverse shall be enabled.

Detailed Configurations

Configure device A:

# Enable the PPP link protocol on the interface and configure the interface IP address; Enable TRIP and split horizon with poisoned reverse.

```
interface Serial 0/0
encapsulation ppp
ip address 192.168.12.1 255.255.255.0
ip rip triggered
ip rip split-horizon poisoned-reverse
```

# Configure the RIP routing protocol.

```
router rip
```

```
network 192.168.12.0
```

Configure device B:

# Enable the PPP link protocol on the interface and configure the interface address; Enable TRIP and split horizon with poisoned reverse.

```
interface Serial 0/0
encapsulation ppp
ip address 192.168.12.2 255.255.255.0
ip rip triggered
ip rip split-horizon poisoned-reverse
```

# Configure RIP routing protocol

```
router rip
network 192.168.12.0
```

Configuring RIPng

Overview

Similar to RIP, RIPng is a distance-vector routing protocol using hop count as the routing metric. RIPng is an interior gateway protocol applicable to small- and medium-sized networks. RIPng is the necessary extension of RIP to address the routing requirements of IPv6. Therefore, RIPng and RIP have basic working principles in common. The main differences rise from the format of their address and packet. Based on IPv6, RIPng supports and uses the multicast group address of FF02::9 for update messages. Security authentication used in RIP is also cancelled. Instead, RIPng enables security authentication by the security mechanism of IPv6. 521 port is used. Packet format, mask and maximum packet length are all different. Please refer to RFC2080 and RFC2081 for details. Given the difference from RIP, the corresponding CLI commands are also lesser.

RIPng Configuration List

- Creating RIPng routing process (Required)
- Enabling RIPng on the interface (Required)
- Adjusting RIPng timer (Optional)
- Configuring Split Horizon (Optional)
- Configuring default metric for redustribution
- Adjusting interface metric
- Configuring the advertisement default route on the interface
- Configuring passive interface
- Configuring RIPng route filtering
- Showing RIPng configuration

Please refer to the chapter of “Protocol independent Configuration” for the configuration of route redistribution.

The default configurations of RIPng are given below:

Function	Default setting
Network interface	The default metric-offset is 1.
Redistribution	The route redistribution is disabled by default. If enabled: ■ All sub-routes of the routing process are redistributed. ■ The metric of redistributed route is the default metric.
Split horizon	Enabled
Poisoned reverse	Disabled

Timer	By default: ■ Update time is 30 seconds ■ Invalid time is 180 second ■ Clearing time is 120 seconds
Default metric	Redistribute the metric used by the routes of other protocols，1 by default.
Administrative distance	120

Creating RIPng Routing Process

In order to run RIPng routing protocol, the routing device first needs to create the RIPng routing process and define the network or interface address associated with RIPng routing process.

To create RIPng routing process, input the following command in the global configuration mode:

Command	Function
Ruijie(config)# **ipv6 router rip**	Create RIPng routing process.

Enabling RIPng on the Interface

To enable RIPng on the interface, input the following command in the interface configuration mode:

Command	Function
Ruijie(config-if)# **ipv6 rip enable**	Enable RIPng on the interface.

In the following example, enable RIPng on ethernet 0/0. Routes in the range of 2001:db8:6::/64 are converged into 2001:db8:6::/64 for advertisement.

```
Ruijie(config)# interface ethernet 0/0
Ruijie(config-if)# ipv6 address 2001:db8:6::1/64
Ruijie(config-if)# ipv6 rip enable
```

Note

Different from RIP, this command enables RIPng on the interface directly without configuring Network command.

Adjusting RIPng Timer

RIPng provides the feature of timer adjustment. You can adjust the timer according to the physical circumstances of the network, so that the RIPng routing protocol can run better. The following timers can be adjusted:

- Route update time (second): defines the interval by which the routing device will send the route update message;

- Route invalid time (second): defines the time by which the route in the routing table becomes invalid upon no update;
- Route clearing time (second): upon expiration of this time, the route will be removed from the routing table.
- By adjusting the aforementioned times, the convergence time and failure recovery time of routing protocol can be accelerated. To adjust RIPng timer, input the following command in the RIPng routing process configuration mode:

Command	Function
Ruijie(config-router)# **times** *update invalid garbage-collection*	Adjust RIPng times.

The following example adjusts the values of three RIPng times:

```
Ruijie(config-router)# timers 10 30 90
```

Caution

Consistency of RIPng times is mandatory for routing devices in the same network.

Configuring Split Horizon

When multiple routing devices are linked to the IP broadcast network and running distance-vector routing protocol, it is necessary to adopt the mechanism of split horizon to avoid loop. Split horizon can prevent routing device from sending routing information to the port from which such routing information was learned. Such mechanism optimizes the routing information exchange between multiple routing devices.

However, for non-broadcast multiple-access network (such as frame relay, X.25 network), split horizon may cause disable certain routing devices from learning all routing information. In such a case, the split horizon will need to be disabled.

To disable or enable split horizon, input the following command in the routing process configuration mode:

Command	Function
Ruijie(config-router)# **no split-horizon**	Disable split horizon.
Ruijie(config-router)# **split-horizon**	Enable split horizon.

By default, split horizon is enabled on all RIPng ports.

Note

The current version can only support split horizon configuration in RIPng routing process, i.e., this command will be applied to all RIPng interfaces. The subsequent RGOS software will realize interfaced-based split horizon configuration.

Different from split horizon, when poison reverse is enabled, the routing device will advertise certain route information from the interface from which such route information was learned. Just set the corresponding metric to infinity (16).

To enable or disable poison reverse, input the following command in the routing process configuration mode:

Command	Function
Ruijie(config-router)#**split-horizon poisoned-reverse**	Enable split horizon with poison reverse
Ruijie(config-router)#**no split-horizon poisoned-reverse**	Disable split horizon with poison reverse

Caution Enabling poison reverse will consume considerable bandwidth.

Configuring Default Metric for Redistribution

When a protocol redistributes routes of other protocols, you need to configure the metric for such redistribution for metric varies by protocols. The default metric of RIPng is 1.

To define the default RIPng metric during the redistribution of other routing protocols, please use the routing process configuration command of "**default-metric"**. Use "**no default-metric"** command to reset the default value to 1.

Command	Function
Ruijie(config-router)# **default-metric** *metric*	The metric of RIPng is set to *metric*
Ruijie(config-router)# **no default-metric**	Reset the default value to 1

Adjusting Interface Metric

Before adding the learned routes into the routing table, you need to add the metric set for the interface to the ones of learned routes. Therefore, you can control the use of routes by configuring the interface metric.

To configure the interface metric, input the following command in the interface configuration mode:

Command	Function
Ruijie(config-if)# **ipv6 rip metric-offset** *value*	Configure interface metric within the scope of 1-16.

The following example sets the metric of ethernet 0/0 to 6:

```
Ruijie(config)# interface ethernet 0/0
Ruijie(config-if)# ipv6 rip metric-offset 6
```

Configuring the Advertisement Default Route on the Interface

To generate an IPv6 default route in the update message of this RIPng process (::/0), input the following command in the interface configuration mode:

Command	Function

Ruijie(config-if)# **ipv6 rip default-informaton originate**	Generate a default route to RIPng on the interface and advertise it with other routes.

To generate an IPv6 default route in the update message of this RIPng process (::/0), and advertise only this default route on this interface, input the following command in the interface configuration mode:

Command	Function
Ruijie(config-if)# **ipv6 rip default-informaton only**	Generate a default route to RIPng on the interface and advertise only this default route.

Configuring Passive Interface

To prevent other routing devices in the local network from learning the routing information sent by the routing device, configure passive interface to disable sending routing update message from this network interface.

To disable sending update messages from an interface, input the following command in the routing process configuration mode:

Command	Function
Ruijie(config-router)# **passive-interface** {**default** | *interface-type interface-num*}	Disable sending update messages from the interface.

Note

When applying the **default** option, all interfaces will be set to passive mode; when applying the **interface** option, the corresponding interface will set to passive mode.

Configuring RIPng Route Filtering

Controlling Route Update Advertisement (RIPng)

To prevent other routing devices in the local network from learning unnecessary routing information, disable the update of specific routes by controlling RIPng route update advertisement.

To disable route update advertisement, input the following command in the routing process configuration mode:

Command	Function
Ruijie(config-router)# **distribute-list prefix-list** *prefix-list-name* **out** [*interface-type interface-name*]	According to the rule of prefix list, enable or disable the advertisement of certain routes.
Ruijie(config-router)# **no distribute-list prefix-list** *prefix-list-name* **out** [*interface-type interface-name*]	Remove the configuration.

In the following example, filtering is only applied to update messages sent from interface eth0, and only update routes included in the prefix-list *outlist* will be sent out.

```
Ruijie(config)# ipv6 router rip
Ruijie(config-router)# distribute-list prefix-list outlist out eth0
```

Controlling Route Update Processing (RIPng)

This feature can be configured to avoid receiving certain routes in the route update message.

To control route update processing, input the following command in the routing process configuration mode:

Command	Function
Ruijie(config-router)# **distribute-list prefix-list** *prefix-list-name* **in** [*interface-type interface-name*]	According to the rule of access list, enable or disable the receipt of certain routes in the route update.
Ruijie(config-router)# **no distribute-list prefix-list** *prefix-list-name* **in** [*interface-type interface-name*]	Remove the configuration.

In the following example, filtering is only applied to update messages received by interface eth0, and only update routes included in the prefix-list *inlist* will be received.

```
Ruijie(config)# ipv6 router rip
Ruijie(config-router)# distribute-list prefix-list inlist in eth0
```

Showing RIPng Configuration

RIPng Debugging Switch

To show the debugging information of RIPng and observe the route processing behaviors of RIPng, input the following command in the privilege configuration mode:

Command	Function
Ruijie# **debug ipv6 rip** [{*interface-type interface-num* **| nsm**}]	Turn on RIPng debugging switch.
Ruijie# **no debug ipv6 rip** [{*interface-type interface-num* **| nsm**}]	Turn off RIPng debugging switch.

Showing RIPng Routing Table

To show RIPng routing table, input the following command in user mode or privileged EXEC mode:

Command	Function
Ruijie# **show ipv6 rip database**	Display RIPng routing table information.

Showing RIPng Routing Process

To display the parameters and various statistical data of RIPng routing process, input the following command in user mode or privilege mode:

Command	Function
Ruijie# **show ipv6 rip**	Display RIPng routing process information.

Showing RIPng Debugging Information

To display the debugging information of RIPng routing process, input the following command in the privileged EXEC mode or global configuration mode:

Command	Function
Ruijie# **show debugging**	Display the debugging information of RIPng routing process.

Configuration Examples

Configuring Default Route Advertisement

Configuration Requirements

There are three devices (see Fig 1 for device connection) running RIPng. The gateway device of Router A advertises the default route to Router B and Router C, with metric being 3. All RIPng interfaces of Router B and Router C are configured to passive mode, so as not to send out RIPng update messages.

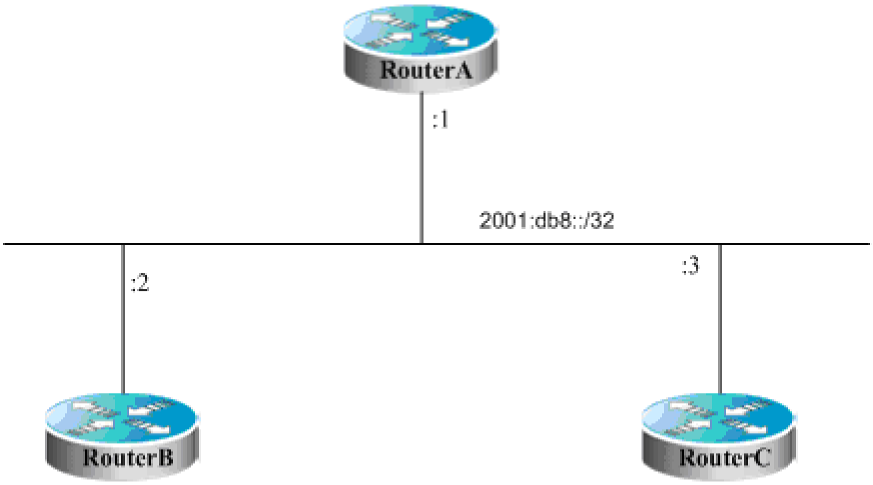


Figure 2 Configuration of default route advertisement

Detailed Configuration of Routing Devices

Router A:

# Configure network interface

```
Ruijie(config)# interface gigabitEthernet 0/1
Ruijie(config-if)# ipv6 enable
Ruijie(config-if)# ipv6 address 2001:db8::1/32
Ruijie(config-if)# ipv6 rip enable
Ruijie(config-if)# ipv6 rip default-information originate
metric 3
# Configure RIPng
Ruijie(config)# ipv6 router rip
Ruijie(config-router)# exit
Router B:
# Configure network interface.
Ruijie(config)# interface gigabitEthernet 0/1
Ruijie(config-if)# ipv6 enable
Ruijie(config-if)# ipv6 address 2001:db8::2/32
Ruijie(config-if)# ipv6 rip enable
# Configure RIPng.
Ruijie(config)# ipv6 router rip
Ruijie(config-router)# passive-interface default
Ruijie(config-router)# exit
Router C:
# Configure network interface.
Ruijie(config)# interface gigabitEthernet 0/1
Ruijie(config-if)# ipv6 enable
Ruijie(config-if)# ipv6 address 2001:db8::3/32
Ruijie(config-if)# ipv6 rip enable
# Configure RIPng.
Ruijie(config)# ipv6 router rip
Ruijie(config-router)# passive-interface default
Ruijie(config-router)# exit
```

Redistribution Configuration

Configuration Requirements

There are three devices (see Fig 1 for device connection). Router A runs RIPng; Router C runs BGP and introduces static routes; Router B needs to redistribute the static routes redistributed by Router C to RIPng domain.

In order to meet such requirements, we can configure the specified community attribute for static routes redistributed to BGP on Router C, while Router B can redistribute BGP routes with specified community attribute to the RIPng domain.

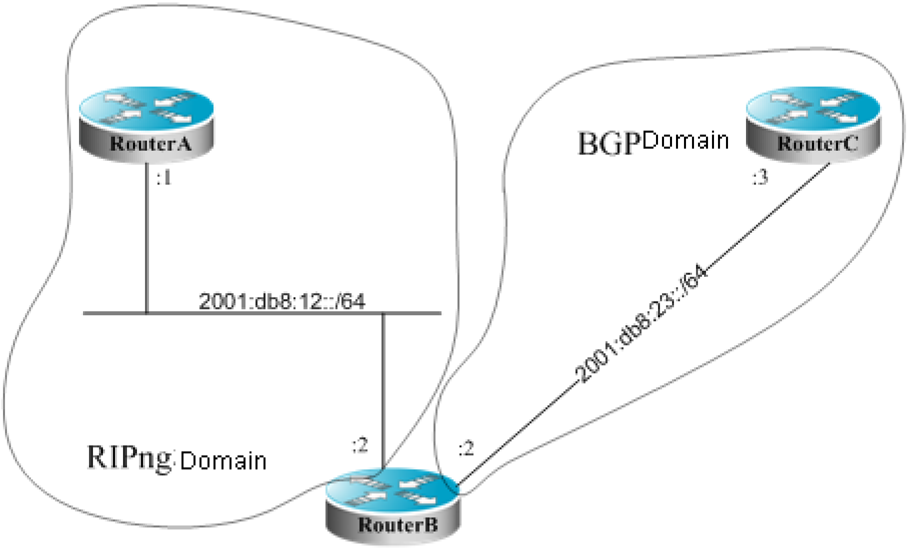


Figure 3 Redistribution configuration

Detailed Configuration of Routing Devices

Router A:

```
# Configure network interface.
Ruijie(config)# interface gigabitEthernet 0/1
Ruijie(config-if)# ipv6 enable
Ruijie(config-if)# ipv6 address 2001:db8:12::1/64
Ruijie(config-if)# ipv6 rip enable
# Configure RIPng.
Ruijie(config)# ipv6 router rip
Ruijie(config-router)# exit
Router B:
# Configure network interface.
Ruijie(config)# interface gigabitEthernet 0/1
Ruijie(config-if)# ipv6 enable
Ruijie(config-if)# ipv6 address 2001:db8:12::2/64
Ruijie(config-if)# ipv6 rip enable
Ruijie(config)# interface gigabitEthernet 0/2
Ruijie(config-if)# ipv6 enable
Ruijie(config-if)# ipv6 address 2001:db8:23::2/64
# Configure RIPng.
Ruijie(config)# ipv6 router rip
Ruijie(config-router)# redistribute bgp route-map riprm
Ruijie(config-router)# exit
# Configure BGP.
Ruijie(config)# router bgp 2
Ruijie(config-router)# neighbor 2001:db8:23::3 remote-as 3
Ruijie(config-router)# address-family ipv6
Ruijie(config-router-af)# neighbor 2001:db8:23::3 activate
# Configure route map.
```

```
Ruijie(config)# route-map riprm
Ruijie(config-route-map)# match community cl_110
# Define community list.
Ruijie(config)# ip community-list standard cl_110 permit 22:22
Router C:
# Configure network interface.
Ruijie(config)# interface gigabitEthernet 0/1
Ruijie(config-if)# ipv6 enable
Ruijie(config-if)# ipv6 address 2001:db8:23::3/64
# Configure BGP.
Ruijie(config)# router bgp 3
Ruijie(config-router)# neighbor 2001:db8:23::2 remote-as 2
Ruijie(config-router)# address-family ipv6
Ruijie(config-router-af)# redistribute static route-map bgprm
Ruijie(config-router-af)# neighbor 2001:db8:23::2 activate
Ruijie(config-router-af)# neighbor 2001:db8:23::2 send-community
# Configure static route.
Ruijie(config)# ipv6 route 2001:db8:88::/64 null 0
Ruijie(config)# ipv6 route 2001:db8:99::/64 null 0
# Configure route map.
Ruijie(config)# route-map bgprm
Ruijie(config-route-map)# set community 22:22
```

Configuring OSPF

Overview

An open shortest path first (OSPF) routing protocol is an internal gateway routing protocol based on link status developed by the IETF OSPF work group It is designed for the IP environment and directly runs on the IP layer. With the protocol number being 89, this routing protocol exchanges OSPF packets in a manner of multicast by using the multicast address 224.0.0.5 (for all OSPF routers) and 224.0.0.6 (for specified routers).

The link status algorithm is an algorithm totally different from the Huffman vector algorithm (distance vector algorithm). The traditional RIP routing protocol uses the Huffman vector algorithm, while the OSPF routing protocol uses the link status algorithm. Compared with the RIP routing protocol, the OSPF routing protocol uses a different algorithm and introduces new concepts such as route update authentication, VLSMs, and route aggregation. Even if the RIPv2 has been improved greatly and also supports the features such as route update authentication and VLSM, the RIP routing protocol still has the following fatal weaknesses: 1) Slow convergence; 2) Limited network scale (with the maximum number of hops counting less than 16). The OSPF routing protocol overcomes these weaknesses, enabling the IGP protocol to be used in large and complicated network environments.

The OSPF routing protocol establishes and calculates the shortest path to each target network by using this complicated link status algorithm. Brief information about how the link status algorithm works is as follows:

- In the initialization stage, a router generates a link status notification that contains all link status of its own.
- All routers exchange the link status information in the multicast way. Upon receiving a link status update packet, each router copies the packet into the local database and then transmits the packet to other routers.
- After every router has a complete link status database, a router uses the Dijkstra algorithm to calculate the shortest path trees to all target networks. The results include the target network, next-hop address, and cost, which are the key parts of an IP routing table.

When there is no link cost or network change, the OSPF is inactive. When any changes occur on the network, the OSPF advertises the link status changes of only the changed links. The routers involved in the changes will run the Dijkstra algorithm again to generate new shortest path trees.

A group of routers running the OSPF routing protocol form the autonomous system of the OSPF routing area. An autonomous system consists of all the routers that are controlled and managed by one organization. Within the autonomous system, only one IGP routing protocol is run. However, between multiple autonomous systems, the BGP routing protocol is used to exchange routing information. Different autonomous area systems may use the same IGP routing protocol. Every autonomous system needs to request the related organization for the autonomous system number to connect to the Internet.

When the OSPF routing area is large, the hierarchical structure can be used. In other words, the OSPF routing area can be divided into several areas, which are connected via a backbone area. Every non-backbone area must be directly connected to the backbone area.

There are three roles for the routers in the OSPF routing area based on their deployment positions:

1) Area internal router: All interface networks of this router belong to a same area.

2) Area border router (ABR): The interface network of this router belongs at least to two areas, one of which must be the backbone area.

3) Autonomous system boundary router (ASBR): It is the router through which routes are exchanged between the OSPF route area and the external route area.

Ruijie products use the OSPF by fully complying with the OSPFv2 defined in RFC 2328. The main features are described as follows:

- Support multiple OSPF processes.
- Support the VRF. You can run the OSPF routing protocol based on different VRFs.
- Support the definition of the stubby area.
- Support route redistribution of static routes, directly-connected routes, dynamic routes, and the routing information among dynamic routing protocols such as RIP and BGP.
- Support plain-text or MD5 authentication between neighbors.
- Support virtual links.
- Support VLSMs.
- Support area division.
- Support the not so stubby area (NSSA) feature, as defined in RFC 3101.
- Support the graceful restart feature, as defined in RFC 3623.

☑ Ruijie products do not support the following functions now: OSPF line support on demand, as defined in RFC 1793; OSPF fast convergence.

Configuration Task List

The configuration of OSPF should be cooperated with various routers, including internal routers, area border routers, and autonomous system boundary routers. When no configuration is performed on routers, default parameters are used. In this case, packets are sent and received without authentication, and the interface does not belong to any area of an autonomous system. When changing the default parameters, you must ensure that the routers have the same configurations.

To configure an OSPF routing protocol, you must perform the following tasks. Among these tasks, creating the OSPF routing process is mandatory. Other tasks may be optional or mandatory in particular applications. The detailed tasks to configure the OSPF routing protocol are described as follows:

- Creating an OSPF routing process (mandatory)
- Configuring OSPF interface parameters (optional)
- Configuring the OSPF used on different physical networks (optional)
- Configuring OSPF area parameters (optional)
- Configuring an OSPF NSSA (optional)
- Configuring OSPF route aggregation (optional)
- Creating a virtual link (optional)
- Generating a default route (optional)
- Using the loopback interface address as the router ID (optional)
- Changing the default OSPF management distance (optional)
- Configuring the route calculation timer (optional)
- Configuring the link status advertisement (LSA) group pacing timer (optional)

- Configuring the cost for the OSPF interface (optional)
- Configuring an OSPF stub router (optional)
- Configuring whether to perform the MTU check on an interface (optional)
- Disabling an Interface to send the OSPF packets (optional)
- Configuring whether to perform the source address check (optional)
- Configuring the OSPF fast convergence function (optional)
- Configuring the OSPF capacity protection function (optional)
- Configuring the OSPF network management function(optional)
- Configuring the OSPF GR function (optional)
- Configuring the OSPF BFD function (optional)
- Configuring the OSPF VPN function (optional)
- Monitoring and maintaining the OSPF

For the configuration information about the following topics, see related sections in *Configuring Protocol-Independent Information*.

- Filtering the routing information
- Redistributing routes

Default OSPF configurations are described as follows:

Feature	Default Setting
Interface parameters	Interface metric: Not preset. LSA retransmission interval: 5 seconds. LSA transmission delay: 1 second. Interval for transmitting Hello packets: 10 seconds (30 seconds for non-broadcast networks) Failure time of adjacent routers: 4 times the interval for transmitting the Hello packets. Fast Hello: Disabled. Priority: 1. Authentication type: 0 (No authentication). Authentication password: None.
Area	Authentication type: 0 (No authentication). Default metric of aggregated routes to a Stub or NSSA area: 1. Inter-area aggregation scope: Undefined. Stub area: Undefined. NSSA: Undefined. Translator for translating Type-7 LSAs to Type-5 LSAs: Alternative. Interval of stabilizing the translation from Type-7 LSAs to Type-5 LSAs: 40 seconds.
Virtual links	No virtual link is defined. The default parameters of the virtual link are as follows: LSA retransmission interval: 5 seconds. LSA transmission delay: 1 second. Interval for transmitting Hello packets: 10 seconds. Failure time of adjacent routers: 4 times the interval for transmitting the

Feature	Default Setting
	Hello packets. Fast Hello: Disabled. Authentication type: No authentication. Authentication password: None.
Automatic cost calculation	Enabled. Default value automatically calculated is 100 Mbit/s.
Default route generation	Disabled. The default metric is 1 and the type is type-2 if enabled.
Default metric (Default metric)	Default metric used to redistribute other routing protocols
Management distance	Intra-area routing information: 110 Inter-area routing information: 110 External routing information: 110
Database filter	Disabled. All interfaces can receive the status update information (LSA).
Neighbor change log	Enabled.
Neighbor	N/A
Neighbor database filter	Disabled. All output LSAs are sent to all neighbors.
Network area (network area)	N/A
Router ID	Undefined. The OSPF protocol does not run on a router by default.
External route aggregation (summary-address)	Undefined.
Changing time of the status update information	240 seconds
Shortest path first (SPF) calculation timer	The time delay between the time for receiving information about topology changes and the next time for invoking the SPF calculation: 1000 milliseconds. The minimum interval between two calculating operations using the SPF algorithm: 5000 milliseconds. The maximum interval between two calculating operations using the SPF algorithm: 10000 milliseconds.
Optimal path rule used to calculate the external routes	Rules defined in RFC1583
OSPF stub router	Disabled.
OSPF two-way maintenance	Enabled.
Sending LSA packet updates	Time interval for sending data packets: 40 milliseconds. Number of LS-UPD packets in each data packet: 10.
OSPF overflow	Enter the overflow state when the memory lacks.
OSPF GR	GR restarter: Disabled. GR helper: Enabled.
OSPFv2 MIB binding	OSPFv2 process with the smallest process number
OSPFv2 TRAP sending	Disabled.

Creating an OSPF Routing Process

You can create an OSPF routing process and define the range of the IP addresses associated with the OSPF routing process and the OSPF area to which these IP addresses belong. The OSPF routing process only sends and receives the OSPF packets at the interface within the IP address range and advertises the link status of the interface to external routers.

Use the following commands to create the OSPF routing process.

Command	Function
Ruijie # **configure terminal**	Enters global configuration mode.
Ruijie (config)# **ip routing**	Enables the IP routing function (if disabled).
Ruijie (config)# **router ospf** [*process_id* [**vrf** *vrf-name*]]	Enables the OSPF and enters OSPF configuration mode.
Ruijie (config-router)# **network** *address wildcard-mask* **area** *area-id*	Defines an IP address range for an area.
Ruijie (config-router)# end	Returns to privileged EXEC mode.
Ruijie # show ip protocols	Displays the routing protocol that is running currently.
Ruijie # write	Saves the configurations.

Note

You can use the parameter *vrf vrf-name* to specify the VRF which the OSPF belongs to. If you do not specify this parameter when creating the OSPF routing process, the default VRF is used. For the **network** command, 32 bit wildcards are opposed to the mask. The value 1 indicates that the bit is not compared, and the value 0 indicates that the bit is compared. However, if you configure the command with masks, Ruijie products automatically translate the masks into bit wildcards. An interface belongs to the specific area as long as the address of the interface is within the IP address range defined in the **network** command. When the address of an interface is within more than one IP address ranges defined in the **network** command of multiple OSPF processes, the OSPF process that the interface involves in is determined based on the optimal mapping.

To disable the OSPF protocol, use the **no router ospf** *process-id* command. The following example shows how to enable the OSPF protocol:

```
Ruijie (config)# router ospf 1
Ruijie (config-router)# network 192.168.0.0 255.255.255.0 area 0
Ruijie (config-router)# end
```

Configuring OSPF Interface Parameters

You are allowed to change some particular interface parameters and configure the interface parameters on demand. It should be noted that some parameters must match those of the adjacent router of the interface. These parameters are set by using the **ip ospf hello-interval**, **ip ospf dead-interval**, **ip ospf authentication**, **ip ospf authentication-key**, and **ip ospf message-digest-key** commands. When you use these commands, make sure that the adjacent routers have the same configurations.

Use the following commands to configure the OSPF interface parameters in interface configuration mode.

Command	Function
Ruijie# **configure** terminal	Enters global configuration mode.

Command	Function
Ruijie(config)# **ip routing**	Enables the IP routing function (if disabled).
Ruijie(config)# **interface** *interface-id*	Enters interface configuration mode.
Ruijie(config-if)# **ip ospf cost** *cost-value*	(Optional) Defines the cost value for the interface.
Ruijie(config-if)# **ip ospf retransmit-interval** *seconds*	(Optional) Sets the link status retransmission interval.
Ruijie(config-if)# **ip ospf transmit-delay** *seconds*	(Optional) Sets the transmission delay for the link status update packets.
Ruijie(config-if)# **ip ospf hello-interval** *seconds*	(Optional) Sets the interval for sending the Hello packets, which must be same for all the nodes of the entire network.
Ruijie(config-if)# **ip ospf dead-interval** *seconds*	(Optional) Sets the dead interval for the adjacent routers, which must be same for all the nodes of the entire network.
Ruijie(config-if)# **ip ospf priority** *number*	(Optional) Priority, used to select the dispatched routers (DR) and backup dispatched routers (BDR).
Ruijie(config-if)# **ip ospf authentication [message-digest | null]**	(Optional) Sets the authentication type on the interface.
Ruijie(config-if)# **ip ospf authentication-key [0|7]** *key*	(Optional) Configures the text authentication key on the interface.
Ruijie(config-if)# **ip ospf message-digest-key *keyid* md5 [0|7**] *key*	(Optional) Configures the key for the MD5 authentication on the interface.
Ruijie (config-if)# **ip ospf database-filter all out**	(Optional) Prevents the interface from flooding the link status update packets. By default, the OSPF floods the LSA information over all interfaces in the same area except the interface on which the LSA information is received.
Ruijie (config-if)# **end**	Returns to privileged EXEC mode.
Ruijie # show **ip ospf interface** [*interface-id*]	Displays the OSPF interface information.
Ruijie # **write**	(Optional) Saves the configurations.

To restore the default value, use the **no** form of the above commands.

Configuring the OSPF Used on Different Physical Networks

According to the transmission features of different media, the networks are classified into three types according to the OSPF protocol:

- Broadcast network (Ethernet, token network, and FDDI)
- Non-broadcast network (frame relay, X.25)
- Point-to-point network (HDLC, PPP, and SLIP)

The non-broadcast networks include two types of networks according to the operation modes of the OSPF:

1) Non-broadcast multi-access (NBMA) network: The NBMA network requires direct communication for all interconnected routers. Only fully meshed networks can meet this requirement. If the SVC (for example, X.25) networking is used, this requirement can be met. However, it is difficult to use the PVC (for example, frame relay)

networking to meet this requirement. The operation of the OSPF on the NBMA network is similar to that on the broadcast network. That is, a designated router must be specified to advertise the link status on the NBMA network.

2) Point-to-multipoint network: If the network is not a fully meshed non-broadcast network, you need to set the network type of the interface to the point-to-multipoint network type according to the OSPF routing protocol. In a point-to-multipoint network, the connections between all routers are treated as point-to-point links according to the OSPF routing protocol, so you do not need to specify the designated router.

Whatever the default network type of the interface is, you can set it to the broadcast network type. For example, you can set the non-broadcast multi-access network (frame relay, X.25) to a broadcast network. The step to configure the neighbor routers can be omitted during the OSPF routing process configuration. By using the **X.25 map** and **Frame-relay map** commands, you can enable the broadcast function on the X.25 and frame relay networks. In this case, the X.25 and frame relay networks are treated as the broadcast networks.

The point-to-multipoint network interface can be seen as the marked point-to-point interface of one or more neighbors. When the network type of is configured to the point-to-multipoint network type according to the OSPF routing protocol, multiple host routes are generated. Compared with the NBMA network, the point-to-multipoint network has the following advantages:

- Easy configuration without configuring the neighbors or specifying the designated router.
- Low cost without requiring fully meshed topology

Use the following command to configure the network type in interface configuration mode.

Command	Function
Ruijie(config-if)# **ip ospf network** {**broadcast** | **non-broadcast** | **point-to-point** | {**point-to-multipoint** [**non-broadcast**]}}	Configures the OSPF network type.

For different link encapsulation types, the default network types described as follows:

- Point-to-point network type:

For PPP, SLIP, frame relay point-to-point subinterface, and X.25 point-to-point subinterface encapsulation

- NBMA (non-broadcast) network type:

For frame relay, X.25 encapsulation (except the point-to-point subinterface)

- Broadcast network type:

For Ethernet encapsulation

- The point-to-multipoint network type has no default.

It should be noted that the network type should be consistent at both sides. Otherwise, exceptions occur. For instance, the neighbor is Full and the routing calculation is incorrect.

Configuring a Point-to-multipoint Broadcast Network

When routers are interconnected using the X.25 and frame relay networks, if the network is not a fully meshed network or you do not want to specify the designated router, you can set the network type of the OSPF interface to the point-to-multipoint type. Since the links are treated as point-to-point links on the point-to-multipoint network, multiple host routes are generated. In addition, all neighbors have the same cost value on the point-to-multiple network. If you want to enable different neighbors to have different cost values, you can set the cost by using the **neighbor** command.

Use the following commands to configure the point-to-multipoint network type in interface configuration mode.

Command	**Function**
Ruijie(config-if)# **ip ospf network point-to-multipoint**	Sets the broadcast network type for an interface to point-to-multipoint.
Ruijie(config-if)# **exit**	Returns to global configuration mode.
Ruijie(config)# **router ospf *1***	Enters routing process configuration mode.
Ruijie(config-router)# **neighbor ip-address cost** *cost*	(Optional). Specifies the cost of the neighbor router.

Note

Although the OSPF point-to-multipoint network is a non-broadcast network, the non-broadcast network is allowed to have the broadcast capability by manual configuration or self-learning according to the frame relay and X.25 mapping. Therefore, you do not need to specify neighbors when configuring the point-to-multipoint network.

Configuring a Non-broadcast Network

When the OSPF routing protocol works on a non-broadcast network, you can set the network type to the NBMA or point-to-multipoint non-broadcast type. Since a non-broadcast network do not have the broadcast capability and cannot dynamically discover neighbors, you must manually configure neighbors for the non-broadcast network when the OSPF routing protocol is used.

Set the network type as the NBMA type in the following conditions:

- When a non-broadcast network has the fully meshed topology;
- When a broadcast network is configured as the NBMA network type to reduce the generation of the broadcast packets, save the network bandwidth, and avoid some arbitrary reception and transmission of routers. During the configuration of the NBMA network, you must specify neighbors and the designated router. Therefore, you must configure the priorities for the routers. It is more possible for the route with a higher priority to be specifies as the designated router.

Use the following commands to set the network type to the NBMA type in interface configuration mode.

Command	**Function**
Ruijie (config-if)# **ip ospf network non-broadcast**	Specifies the network type of the interface to NBMA.
Ruijie (config-if)# **exit**	Returns to global configuration mode.
Ruijie (config)# **router ospf** *1*	Enters routing process configuration mode.
Ruijie(config-router)# **neighbor** *ip-address* [*priority number*] [*poll-interval seconds*]	Specifies the neighbor, its priority, and polling interval of Hello packets.

The best solution is to set the network where the OSPF is used to the point-to-multipoint non-broadcast network when you cannot make sure whether any two routers on a non-broadcast network are reachable directly.

All neighbors on the point-to-multipoint broadcast or non-broadcast network have the same cost value which is configured by using the **ip ospf cost** command. However, the bandwidth of each neighbor may be different, so the cost is different. You can specify the cost for each neighbor by using the **neighbor** command. However, this only applies to the interface used on the point-to-multipoint type (broadcast or non-broadcast) network.

Use the following commands to set the type of the interface as the point-to-multipoint type on a non-broadcast network in interface configuration mode.

Command	Function
Ruijie (config-if)# **ip ospf network point-to-multipoint non-broadcast**	Specifies the network type of the interface to be the point-to-multipoint non-broadcast network type.
Ruijie (config-if)# **exit**	Returns to global configuration mode.
Ruijie (config)# **router ospf** *1*	Enters routing process configuration mode.
Ruijie(config-router)# **neighbor** *ip-address* [*cost number*]	Specifies the neighbor and the cost to the neighbor.

Pay attention to step 4. If you have not specified the cost for the neighbor, the cost referenced in the **ip ospf cost** command in interface configuration mode is used.

Configuring the Broadcast Network Type

It is necessary to specify the designated router (DR) and backup designated router (BDR) for the OSPF broadcast network. The DR advertises the link status of this network to external devices. All routers keep the neighbor relationship with each another and only the adjacent relationship with the designated router and backup designated router. That is to say, each router only exchanges the link status packets with the designated router and backup designated router. Then the designated router advertises the link status information to all other routers. As a result, each router can store a consistent link status database.

You can control the specifying result of the designated router by configuring the OSPF priority. This parameter does not take effect immediately until the new round for specifying the designated router. The new round for specifying the designated router occurs only when the OSPF neighbors do not receive the Hello packets from the designated router within the specified time and judge that the DR is down.

Use the following commands to configure the broadcast network type in interface configuration mode.

Command	Function
Ruijie(config-if)# **ip ospf network broadcast**	Specifies the type of the interface to be the broadcast network type.
Ruijie(config-if)# **ip ospf priority** *priority*	(Optional) Specifies the priority of the interface.

Configuring OSPF Area Parameters

To configure area authentication, stub area, and default route summary cost, you need to configure area commands.

The area authentication is used to prevent from learning non-authenticated and invalid routes and advertising valid routes to non-authenticated routers. In a broadcast network, the area authentication can also prevent the non-authenticated routers from becoming the designated routers, therefore improving the stability and intrusion prevention capability of the routing system.

When an area is an OSPF leaf area, that is, the area neither acts as a transit area nor injects external routes to the OSPF area, you can configure the area as a stub area. The routers in a stub area can only learn about three routes: 1) Routes in the stub area, 2) Routers in other areas, and 3) Default routes advertised by the stub area border router. There are few external routes, so the size of the routing tables of the routers in the stub area is small, and fewer router resources are used. The routers in the stub area may be low- or middle-level routers. To further reduce the number of the LSAs sent to the stub area, you can configure the area as a totally stub area (configured with the **no-summary** option). The routers in the totally stub area can learn two types of routes: 1) Routes in the totally stub area; 2) Default routes advertised by the border router in the totally stub area. After the totally stub area is configured, the router resources occupied by the OSPF are minimized, therefore improving the network transmission efficiency.

If the routers in a stub area can learn multiple default routes, you need to set the costs for these default routes by using the **area default-cost** command, so that the routers in the stub area use the specified default routes with priority.

Pay attention to the following aspects when configuring a stub area:

- A backbone area cannot be configured as a stub area, and the stub area cannot be used as the transmission area of virtual links.
- There is no ASBR in the stub area. In other words, the routes outside an autonomous system cannot be transmitted in this area.
- To set an area as the STUB area, configure all routers in this area with the same attribute.

Use the **no area *area-id*** command to remove the configurations of the specified OSPF area and delete the area, including deleting the area-based configuration commands such as **area authentication**, **area default-cost**, **area filter-list**, **area stub**, and **area nssa**. However, the user cannot remove the OSPF area configurations in the following circumstances:

3) The user needs to remove all configurations of the backbone area, however, configurations of virtual links exist. In this case, the user can delete the backbone area only when the configurations of virtual links are removed.

4) The corresponding **network area** command exists in any area. In this case, the user can delete the area only when all network segment commands added in this area are removed.

Use the following commands to configure the OSPF area parameters in routing process configuration mode.

Command	Function
Ruijie (config-router)#**area** *area-id* **authentication**	Sets plain-text authentication for the area.
Ruijie (config-router)#**area** *area-id* **authentication message-digest**	Sets MD5 authentication for the area.
Ruijie (config-router)#area *area-id* **stub** [**no-summary**]	Sets the area as a stubby area. **no-summary**: Sets the area as a stubby area to prevent the ABR in a stub area from sending summary-LSA information to the stub area.
Ruijie (config-router)#**area** *area-id* **default-cost** *cost*	Configures the cost of the default route sent to the stub area.

Note

When configuring the authentication, you need to configure the authentication parameters on an interface. For more details, see the "Configuring OSPF Interface Parameters" section. You must configure all routers in the area to have the same configuration with the stub area. To configure a totally stub area, you also have to configure the totally stub area parameters on the border routers in the stub area in addition to the basic configuration of the stub area, and you do not need to change the configurations of other routers.

Configuring an OSPF NSSA

The NSSA is an expansion of the OSPF stub area. In the NSSA, the consumption of router resources is reduced by preventing the type-5 LSAs (AS-external-LSA) from flooding to the NSSA. However, unlike the stub area, the NSSA can inject some routing information outside the autonomous system to the OSPF routing area.

Through the route redistribution, the external AS routes (type-7) are allowed to import to the NSSA. These external type-7 LSAs are converted into the type-5 LSAs on an area border router in the NSSA and flooded to the entire autonomous system. In this process, the external routes are aggregated and filtered.

Pay attention to the following aspects when configuring the NSSA:

- A backbone area cannot be configured as an NSSA, and the NSSA cannot be used as the transmission area of virtual links.
- To set an area as the NSSA, configure all routers connected to the NSSA with the NSSA attribute by using the **area nssa** command.

Use the following commands to configure an area as the NSSA area in routing process configuration mode.

Command	Function
Ruijie (config-router)# area *area-id* **nssa** [**no-redistribution**] [**no-summary**] [**default-information-originate[metric** *metric*][**metric-type** [**1** | **2**]]] [**translator** [**stability-interval** *seconds* | **always**]]	(Optional) Defines an NSSA area.
Ruijie (config-router)# **area** *area-id* **default-cost** *cost*	Configures the cost of the default route sent to the NSSA area.

Use the *default-information-originate* parameter to generate the default Type-7 LSA. This option varies slightly between the ARR and ASBR in the NSSA. On the ABR, whether the routing table contains a default route or not, the default Type-7 LSA route is generated. On the ASBR, the default Type-7 LSA route is generated only when the routing table of the ASBR contains a default route.

If the *no-redistribution* parameter is configured on the ASBR, other external routes introduced by using the **redistribute** commands are not allowed to be distributed to the NSSA. This option is usually used when the router in the NSSA is both an ASBR and an ABR. This option can also prevent the external routing information from entering the NSSA.

To further reduce the number of the LSAs sent to the NSSA, you can configure the *no-summary* parameter on the ABR to prevent the ABR from sending the aggregated LSAs (Type-3 LSAs) to the NSSA.

In addition, the *area default-cost* parameter is used on the ABR/ASBR connected to the NSSA. This option is used to configure the cost of the default route sent by the ABR/ASBR to the NSSA. By default, the cost value of this default route is 1.

If two or more than two ABRs exist in an NSSA area, the ABR with the largest ID is selected as the translator to translate the Type-7 LSAs to the Type-5 LSAs by default. You can use the *translator always* parameter to configure the current router as the permanent translator ABR.

If the translator role is acted by other ABRs, the current router keeps the capability within the **stability-interval** time. If the router is not configured as the translator again within this period, after the **stability-interval** time expires, LSAs translated from Type-7 to Type-5 will be removed from the AS.

Note

The Type-5 LSAs aggregated and translated from the Type-7 LSAs are removed immediately after the current router becomes translator-disabled without waiting for the **stability-interval** timeout to prevent routing loop.

In the same NSSA area, it is recommended to configure the *translator always* parameter for only one ABR.

Configuring OSPF Route Aggregation

Configuring the Route Aggregation between Areas

An area border router (ABR) has at least two interfaces that belong to different areas, one of which must be a backbone area. The ABR acts as the pivot in the OSPF routing area. It can advertise the routes of one area to another area. If the network addresses of the routes of this area are continual, the ABR can advertise only one aggregated route to other areas. The route aggregation function between areas greatly reduces the size of the routing table and improves the network efficiency.

Use the following command to configure the route aggregation between areas in routing process configuration mode.

Command	Function
Ruijie (config-router)# **area** *area-id* **range** *ip-address mask* [**advertise** | **not-advertise**] [**cost** *cost*]	Configures the route aggregation between areas.

Note

If the route aggregation is configured, the ABR does not advertise the detailed routes in this area to other areas.

Configuring the External Route Aggregation

When routes are redistributed in other routing processes and imported into the OSPF routing process, every route is advertised to the OSPF-enabled router as a separate link. If the injected routes have continuous IP addresses, an ASBR can advertise only one aggregated route, therefore reducing the size of the routing table significantly.

Use the following command to configure the external route aggregation in routing process configuration mode.

Command	Function
Ruijie (config-router)# **summary-address** *ip-address mask* [**not-advertise** | **tag** *tag-id*]	Configures the external route aggregation.

Controlling the Aggregated Routes to Be Added to the Routing Table

The network range after the route aggregation may exceed the original network range in the routing table. If data are sent to the network beyond the aggregation range, routing loop may incur or load on the router may increase. Therefore, add a discard route to the routing table of the ABR or ASBR to prevent that problem.

Use the following commands to allow or forbid adding the discarded routes to the routing table in routing process configuration mode.

Command	Function
Ruijie (config-router)# **discard-route** {**internal** | **external**}	Allows adding the discarded routes to the routing table.
Ruijie (config-router)# **no discard-route** {**internal** | **external**}	Forbids adding the discarded routes to the routing table.

By default, adding the discarded routes to the routing table is allowed.

Creating a Virtual Link

In an OSPF routing area, the OSPF route update between non-backbone areas is performed by using the backbone area. All non-backbone areas are connected to the backbone area. If the backbone area is disconnected with the non-backbone areas, you need to configure virtual links to connect the non-backbone areas to the backbone area. Otherwise, the network communication fails. Create virtual links for the connections when physical links cannot meet the requirements due to the limitation of the network topology.

A virtual link can be created between two ABRs. The common area that the two ABRs belong to is a transit area. A stub area and NSSA area cannot be used as the transit area. The virtual link can be seen as a logical connection channel established between the two ABRs via the transit area. On both ends of the virtual link deploy ABRs and configuration on both ends must be performed synchronously. The virtual link is identified with the router ID of the peer router. The area that provides the two ends of the virtual link with an internal non-backbone area route is called the transit area, whose number must be specified during the configuration.

The virtual link will be activated after the route in the transit area has been calculated (that is, the route to the peer router). You can see it as a point-to-point link, on which most parameters of the interface, such as the *hello-interval* and *dead-interval* parameters, can be configured like a physical interface.

A logical channel means that multiple routers running the OSPF routing protocol between the two ABRs. The logical channel is only used to forward packets. (Since the destination of the protocol packets are not these routers, the packets are transparent to them and are simply forwarded as common IP packets.) The two ABRs exchange routing information directly, and the synchronization mode in the area is not changed. The routing information means the Type-3 LSAs generated by the ABR.

Use the following command to create a virtual link in routing process configuration mode.

Command	**Function**
Ruijie (config-router)# **area** *area-id* **virtual-link** *router-id* [[**hello-interval** *seconds*]| [**retransmit-interval** *seconds*] |[transmit-delay *seconds*]|[**dead-interval** *seconds*]| [**authentication** [**message-digest** | **null**] |[[**authentication-key** [**0**|**7**] *key* | **message-digest-key** *keyid* **md5** [**0**|**7**] *key*]]]	Creates a virtual link.

Caution

If the autonomous system is divided into more than one area, one of the areas must be the backbone area to which the other areas must be connected directly or logically. Also, the backbone area must be in good connection.

Note

The *router-id* is the ID of an OSPF neighbor router. If you are not sure of the router-id, you can use the **show ip ospf** or **show ip ospf neighbor** command to check it. For information about how to manually configure the router-id, see the "Using the Loopback Interface Address as the Route ID" section.

Generating a Default Route

An ASBR can be forced to generate a default route, which is injected to the OSPF routing area. If a router is forced to generate the default route, it is configured to be the ASBR automatically. However, the ASBR does not automatically generate the default route.

Use the following command to force the ASBR to generate a default route in routing process configuration mode.

Command	**Function**
Ruijie (config-router)# **default-information originate** [**always**] [**metric** *metric-value*] [**metric-type** ***type***-*value*] [**route-map** ***map***-*name*]	Generates a default route.

Note

When the stub area is configured, the ABR generates the default route automatically and advertises the default route to all routers within this stub area.

Using the Loopback Interface Address as the Router ID

In an OSPF routing process, the largest interface IP address is always used as the router ID. If the interface is disabled or the IP address does not exist, the router ID must be calculated again and all the routing information is sent to the neighbors.

If the loopback interface address (local loop address) is configured, then in the routing process, the IP address of the loopback interface is used as the router ID. If there are multiple loopback interfaces, the largest IP address is selected as the router ID. The loopback address always exists, therefore improving the stability of the routing table.

Use the following commands to configure the loopback address in global configuration mode.

Command	Function
Ruijie (config)# **interface loopback** *1*	Creates the loopback interface.
Ruijie (config-if)# **ip address** *ip-address mask*	Configures the loopback IP address.

Note

When the IP address of a common interface is specified as the route identifier in the OSPF routing process, even if the loopback interface is configured, the identifier is not specified once again in the OSPF process.

Changing the Default OSPF Management Distance

The management distance of a route represents the credibility of the route source. The management distance ranges from 0 to 255. The greater this value is, the lower the credibility of the route source is.

The OSPF function supported on Ruijie products supports intra-area, inter-area, and external routes, whose management distances are all 110 by default. A route belongs to a same area is called the intra-area route, a route to another area is called the inter-area route, and a route to another routing area (learned through redistribution) is called the external route.

Use the following command to change the OSPF management distance in routing process configuration mode.

Command	Function
Ruijie(config-router)# **distance** {*distance* | **ospf** {**intra-area** *distance* | **inter-area** *distance* | **external** *distance*}}	Changes the OSPF management distance.

Configuring the Route Calculation Timer

When receiving a notification about route topology changes in the OSPF routing process, the system runs the SPF algorithm for route calculation after a time delay. You can configure this delay and also configure the minimum interval between two SPF calculations.

Use the following command to configure the OSPF route calculation timer in routing process configuration mode.

Command	Function
Ruijie (config-router)# **timers throttle spf** *spf-delay spf-holdtime spf-max-waittime*	Configures the route calculation timer.

Note

The *spf-delay* refers to the delay time from the time when the topology changes to the time when the SPF calculation is performed. The *spf-holdtime* refers to the minimum time interval between two SPF calculations. Later, the time interval of the consecutive SPF calculations shall be at least twice as the last time interval until the time interval reaches the *spf-max-waittime* value*.* If the time interval between two SPF calculations has exceeded the minimum value, then the time interval is recalculated from the *spf-holdtime*.

Normally, reducing the value of *spf-delay* and *spf-holdtime* can speed up the OSPF convergence if the link turbulence occurs occasionally. Increasing the value of the *spf-max-waittime* can avoid the CPU consumption by the OSPF routing process due to the consecutive link turbulence.

For example, timers throttle spf 1000 5,000 100,000

If the topology changes constantly, the time interval for SPF calculations increases in the ascend order as follows when calculated by using the binary exponential backoff algorithm, but this time interval does not exceed the *spf-max-waittime*:

1 second, 6 seconds, 16 seconds, 36 seconds, 76 seconds, 156 seconds, 256 seconds, 256+100 seconds…

Use the following command to configure only the OSPF route calculation delay and hold-time in routing process configuration mode.

Command	Function
Ruijie (config-router)# **timers spf** *spf-delay spf-holdtime*	Configures the route calculation timer in seconds.

Caution

The **timers spf** and **timers throttle spf** commands overwrite each other during the configuration. The latter configured command takes effect. If neither of the two commands are configured, the default value is the value configured in the **timers throttle spf** command.

The **timers throttle spf** command is more powerful than the **timers spf** command. Therefore, you are advised to use the **timers throttle spf** command.

Changing the LSA Group Pacing Timer

Each LSA has its own refresh and aging time. Calculating the refresh and aging time for each LSA respectively consumes lots of CPU. To make full use of the CPU, perform the refresh and aging operations for the LSAs uniformly.

The default refresh and aging time is 4 minutes. You do not need to adjust this parameter often. The optimum group pacing interval is inversely proportional to the number of LSAs that need to be calculated. For example, if approximately 10,000 LSAs exist in the database, decreasing the pacing interval would be better. If only 40 to 100 LSAs exist in the database, increasing the pacing interval to 10 to 20 minutes might be better.

Use the following commands in routing process configuration mode.

Command	Function
Ruijie # **configure terminal**	Enters global configuration mode.
Ruijie (config)# **router ospf** *1*	Enables the OSPF and enters OSPF configuration mode.
Ruijie (config-router)# **timers pacing lsa-group** *seconds*	(Optional) Changes the LSA group pacing.
Ruijie (config-router)# **end**	Returns to privileged EXEC mode.
Ruijie # **show running-config**	Verifies the configurations.

Command	Function
Ruijie # **write**	(Optional) Saves the configurations.

To restore the default value, use the **no timers pacing lsa-group** in global configuration mode.

Configuring the Cost for the OSPF Interface

The OSPF system calculates the destination route based on cost. The route with the least cost is the shortest route. The default route cost is based on network bandwidth. When you configure the OSPF-enabled router, you can set the link cost according to the factors such as link bandwidth, time delay or economic cost. The lower the link cost is, the higher the possibility is for the link to be selected as the route. If route aggregation is enabled, the maximum cost of all the aggregated links are used as the cost of the aggregated information.

Routing configuration includes two steps. First, specify a reference value for the generated cost based on the bandwidth. This value and the interface bandwidth are used to calculate the default cost. Second, set the cost for each interface by using the **ip ospf cost** command. In this case, the default cost takes no effective on the interface. For example, if the default reference value is 100 Mbit/s, and the bandwidth of an Ethernet interface is 10 Mbit/s, the default cost of this interface is 100/10 + 0.5 ≈ 10.

The interface cost is selected in the following way according to the OSPF protocol: The cost of the interface specified by the user has the highest priority. If you have specified the interface cost, the cost is used as the interface cost. If you do not specify the interface cost but the automatic cost generation function is enabled, the automatically calculated value is used. If the automatic cost generation function is disabled, the default value 10 is used.

Use the following commands to perform the configuration.

Command	Function
Ruijie # **configure terminal**	Enters global configuration mode.
Ruijie (config)# **router ospf** *1*	Enters routing protocol configuration mode.
Ruijie(config-router)# **auto-cost reference-bandwidth** *ref-bw*	(Optional) Sets the default cost based on the bandwidth on an interface. The cost value is determined based on the *ref-bw* parameter.
Ruijie (config-router)# **end**	Returns to privileged EXEC mode.
Ruijie # **show ip protocols**	Displays the routing protocol that is running currently.
Ruijie # **write**	(Optional) Saves the configurations.

To remove the setting, use the **no ip ospf cost** and **auto-cost** command.

Configuring an OSPF Stub Router

A router that only forwards packets to its directly-connected links is called a stub router. To prevent low-level routers from handling massive LSAs or to enable the routers to smoothly join/exit a network, you can configure such routers as stub routers. The stub router can advertise its maximum cost so that other routers will not preferentially use this router as a transit node during SPF calculation.

After the **max-metric router-lsa** command is enabled, the metric of non-stub links carried in the Router LSA generated by the router will be used as the maximum value (0xFFFF). After the user removes the setting or the timer expires, the default metric of the links is restored.

By default, after this command is enabled, the ordinary metric of stub links is advertised, namely the cost of the egress. If the *include-stub* parameter is configured, the maximum metric of the stub links is advertised.

If you do not want to transmit the data in an area, use the *summary-lsa* parameter to configure the summary LSA as the maximum metric for the ABR.

If you do not want to transmit the data in an external area, use the *external-lsa* parameter to configure the external LSA as the maximum metric for the ASBR.

The **max-metric router-lsa** command is generally used in the following circumstances:

- Restart the router. After the router is restarted, the IGP protocol is converged more quickly, and other routers may try to forward the data through the restarted router. If the router is still building BGP routing tables and certain BGP routes have not be learned, packets sent to such router will be discarded. In this case, use the on-startup parameter to configure a delay timer, so that the restarted router can act as the transit node after the timer runs out.
- Connect the router to the network without using the router to transmit the packets. If alternative paths exist, the current router will not be used to transmit the packets. If no alternative path exists, the current router will still be used to transmit the packets.
- Gracefully remove the router from the network. By using this command, the current router can advertise a maximum metric value, so that other routers on the network will select the alternative paths to transmit the packets before the router is shut down.

Use the following commands to configure a router to advertise a maximum metric in routing process configuration mode.

Command	Function
Ruijie # **configure terminal**	Enters global configuration mode.
Ruijie (config)# **router ospf** *1*	Enables the OSPF and enters OSPF configuration mode.
Ruijie (config-router)# **max-metric router-lsa** [**external-lsa** [*max-metric-value*]] [**include-stub**] [**on-startup** [*seconds*]] [**summary-lsa** [*max-metric-value*]]	(Optional) Configures the router to advertise a maximum metric.
Ruijie (config-router)# **end**	Returns to privileged EXEC mode.
Ruijie # **show ip protocols**	Displays the routing protocol that is running currently.
Ruijie # **write**	(Optional) Saves the configurations.

Caution

In the earlier versions of the OSPF (RFC 1247 or earlier versions), links with the maximum metric (0xFFFF) in the LSAs are not involved in the SPF calculation, that is, no packet is sent to routers generating these LSAs.

Configuring Whether to Perform the MTU Check on an Interface

When the OSPF system receives database description packets, it will check whether the MTU of a neighbor interface is the same as its own MTU. If the MTU of the interface indicated in the received database description packets is greater than that of the receiving interface, the adjacency relationship cannot be established. In this case, you can disable the MTU check function.

Use the following command to disable the MTU check on an interface in interface configuration mode.

Command	Function
Ruijie (config-if)# **ip ospf mtu-ignore**	Disables the MTU check on the interface when the interface receives the database description packets.

The MTU check on an interface is disabled by default.

Disabling an Interface to Send the OSPF Packets

To prevent other routers on the network from dynamically learning the routing information of the local router, you can use the **passive-interface** command to set the specified network interface of the local router as a passive interface to prevent the local router from sending the OSPF packets.

Use the following commands to configure an interface as a passive interface in privileged EXEC mode.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **router ospf** *1*	Enters routing protocol configuration mode.
Ruijie(config-router)# **passive-interface** *interface-name*	(Optional) Sets the specified interface as a passive interface.
Ruijie(config-router)# **passive-interface default**	(Optional) Sets all network interfaces as the passive interfaces
Ruijie(config-router)# **end**	Returns to privileged EXEC mode.
Ruijie# **write**	Saves the configurations.

By default, all interfaces are allowed to receive and send the OSPF packets. To re-enable the network interface to send the routing information, use the **no passive-interface *interface-id*** command. To re-enable all network interfaces, use the keyword **default.**

Configuring Whether to Perform the Source Address Check

According to the OSPF requirements, the source address of the received packets must be in the same network segment with the address of the interface that receives the packets. However, for a point-to-point link, the addresses of the two link ends are configured independently, so the addresses are not required to be in the same network segment. In the negotiation process of a point-to-point link, the address information about the peer end is advertised. Therefore, the OSPF system will check whether the source address of the packets is the address advertised by the peer end during the negotiation. If the two addresses are not consistent, the system will treat the packets as unauthorized packets and discard the packets. The negotiated address may be shielded in some applications. In this case, disable this source address check function to establish the OSPF adjacency relationship properly. In particular, this function is disabled on the unnumbered interface all the time.

Use the following command to configure whether to perform the source address check on a point-to-point link in interface configuration mode.

Command	Function
Ruijie (config-if)# **ip ospf source-check-ignore**	Disables the source address check on the point-to-point link.

The source address check on a point-to-point link is enabled by default.

Configuring the OSPF Fast Convergence Function

Configuring the OSPF Fast Hello

The OSPF Fast Hello function facilitates fast discovery of OSPF neighbors and supports quick detection of lost OSPF neighbors. The OSPF Fast Hello function is enabled by specifying the **minimal** and **hello-multiplier** keywords and the *multiplier* parameter. The **Minimal** keyword is used to set the dead interval to 1 second, and the **hello-multiplier** keyword is used to configure the times for sending Hello packets during the dead interval, therefore the interval for sending the Hello packets is reduced to less than 1 second.

When the Fast Hello function is enabled on the interface, the **Hello interval** field for the interface sending the Hello packets is set to 0. The **Hello interval** field for the interface receiving the Hello packets is ignored.

No matter whether the Fast Hello function is enabled or not, the dead interval must be consistent on a same network segment. However, the *hello-multiplier* parameter is not required to be consistent on a same network segment as long as at least one Hello packet is received within the dead interval.

Use the following commands to configure the Fast Hello on an interface.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **ip routing**	Enables the IP routing function (if disabled).
Ruijie(config)# **interface** *interface-id*	Enters interface configuration mode.
Ruijie(config-if)# **ip ospf dead-interval minimal hello-multiplie**r *multiplier*	(Optional) Enables the OSPF Fast Hello on the interface.
Ruijie (config-if)# **end**	Returns to privileged EXEC mode.
Ruijie # **show ip ospf** [*process-id*]**interface** [*interface-id*]	Displays the OSPF interface information.
Ruijie # **write**	Saves the configurations.

Use the following commands to configure the Fast Hello on a virtual link.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie (config)# **ip routing**	Enables the IP routing function (if disabled).
Ruijie (config)# **router ospf** *process_id* [**vrf** *vrf-name*]	Enables the OSPF and enters OSPF configuration mode.
Ruijie (config-router)# **area** *area-id* **virtual-link** *router-id* [**dead-interval** | **minimal hello-multiplier** *multiplier*]	Enables the Fast Hello on the virtual link
Ruijie (config-router)# **end**	Returns to privileged EXEC mode.
Ruijie # **write**	Saves the configurations.

Caution

You cannot configure the *dead-interval minimal hello-multiplier* parameter and the *hello-interval* parameter at the same time.

Configuring the OSPF Two-Way Maintenance

On a large scale network, a large number of packets may be received and transmitted, which occupies high CPU and memory resources, therefore causing the delay or drop of certain packets. If the time for processing the Hello packets goes beyond the dead interval, the corresponding adjacent routers will be disconnected. In this case, enable the OSPF two-way maintenance function. If a large number of packets exist on the network, besides the Hello packets, the DD, LSU, LSR and LSAck packets from a certain neighbor can also be used to maintain the two-way adjacency relationship, therefore avoiding the disconnection of neighbors caused by the delay or drop of the Hello packets.

The OSPF two-way maintenance function is enabled by default. Use the following commands to disable the OSPF two-way maintenance function in routing process configuration mode.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **router ospf** *1*	Enters routing protocol configuration mode.
Ruijie(config-router)# **no two-way-maintain**	(Optional) Disables the OSPF two-way maintenance function.
Ruijie(config-router)# **end**	Returns to privileged EXEC mode.
Ruijie# **write**	Saves the configurations.

Configuring the Interval of Receiving the Same LSA

On a broadcast network or in the environment featured by frequent network oscillation, the router may receive the same LSA updates from one or multiple interfaces and different neighbors. If the same LSAs are processed every time, excessive system resources are wasted. According to the OSPF protocol, the same LSAs are considered to be valid after a period of time. The same LSAs received within a short period of time will be ignored. This time interval is the constant MinLSArrival with the value set to 1 second.

Different types of networks have different requirements on the interval for processing LSA changes. The user can configure this parameter according to different network planning and performance requirements to optimize the network.

Use the following commands to configure the interval of receiving the same LSA in routing process configuration mode.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **router ospf** *1*	Enters routing protocol configuration mode.
Ruijie(config-router)# **timers lsa arrival *arrival***-*time*	(Optional) Configures the interval of receiving the same LSA. The default value is 1000 milliseconds.
Ruijie(config-router)# **end**	Returns to privileged EXEC mode.
Ruijie# **write**	Saves the configurations.

Configuring to Send the LSA Packet Updates

To relieve the impacts on network devices caused by the flooding of a large number of update packets, the LSP packet update function is introduced. By specifying the delay interval for the update packets, the LSAs to be flooded during the interval can be collected, so that these LSAs can be sent with the least number of packets. Meanwhile, the CPU can process other tasks and the system performance is optimized.

When a large number of LSAs exist on the network and the router loads excessively, you need to configure the **transmit-time** and **transmit-count** commands properly to control the number of LS-UPD packets flooded on the network. When the load of CPU is low and the network bandwidth is small, you can reduce the transmit-time value and increase the transmit-count value to speed up the network convergence.

Use the following commands to configure the LSA to send the packet updates in routing process configuration mode.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **router ospf** *1*	Enters routing protocol configuration mode.
Ruijie(config-router)# **timers pacing lsa-transmit** *transmit-time transmit-count*	(Optional) Configures the LSA to send the packet updates.
Ruijie(config-router)# **end**	Returns to privileged EXEC mode.
Ruijie# **write**	Saves the configurations.

Configuring the Exponential Backoff Algorithm for Generating LSAs

To prevent multiple events from triggering the same LSAs within a short time, causing frequent LSA updates, and consuming excess CPU resources, you can specify the minimum time interval "MinLSInterval" for generating the LSAs according to the OSPF routing protocol. The default minimum time interval is 5 seconds. During this time interval, the same LSA instances cannot be generated repeatedly, therefore preventing frequent LSA oscillation from causing impacts on the network. However, this configuration slows down the LSA generation speed and fails to advertise the network topology changes immediately.

To quickly respond to the network topology changes and avoid excessively frequent route calculations, use the exponential backoff algorithm to dynamically change the time interval for generating the LSAs. The **timers throttle lsa all** command has three parameters: *delay-time*, *hold-time*, and *max-wait-time*, which allow the system to automatically adjust the time interval for generating the LSAs according to the frequency of the network topology changes. Generally, *delay-time* is set to a small value or 0 to trigger LSA instances immediately when the network topology is comparatively stable. When the network topology changes frequently, the time interval for generating the LSAs increases from the *hold-time* and follows the algorithm of *hold-time* x 2^{n-1}. n refers to the times of changes. With the times for generating LSAs repeatedly increasing, the time interval for generating the LSAs becomes greater and greater until the *max-wait-time* is reached. When the time interval for generating the LSAs is greater than the *max-wait-time*, the *delay-time* for generating the LSAs restores to the initial value.

By default, the initial value is 0 milliseconds, the *hold-time* is 5000 milliseconds, and the *max-wait-time* is 5000 milliseconds. The shortest interval for consecutively generating the same LSA is the *MinLSInterval*, which complies with the rules defined in RFC 2328.

Use the following commands to configure the exponential backoff algorithm for generating the LSAs in routing process configuration mode.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **router ospf** *1*	Enters routing protocol configuration mode.
Ruijie(config-router)# **timers throttle lsa all** *delay-time hold-time max-wait-time*	(Optional) Configures the exponential backoff algorithm for generating the LSAs. By default, the initial value is 0 milliseconds, the *hold-time* is 5000 milliseconds, and the *max-wait-time* is 5000 milliseconds.
Ruijie(config-router)# **end**	Returns to privileged EXEC mode.
Ruijie# **write**	Saves the configurations.

Caution During the configuration, the *hold-time* cannot be less than the *delay-time*, and the *max-wait-time* cannot be less than the *hold-time*.

Configuring the OSPF Capacity Protection Function

When the memory lacks, the OSPF system enters the overflow state. In the overflow state, the OSPF protocol triggers the following operations:

- For the learned LSAs: receive the inter-area and intra-area LSAs, and only receive the external LSAs indicating that the route to the destination address is a specific non-default route.
- For the external LSAs generated by itself: clear the external LSAs except for the LSAs indicating the default routes.
- The incompleteness of route learning and advertisement may lead to the routing loop on the network. The OSPF system generates a default route to the NULL interface to prevent the routing loop. The generated default route exists in the overflow state all the time.

Use the following commands to configure the OSPF router to enter the overflow state when the memory lacks.

Command	Function
Ruijie(config)#**router ospf** *process-id*	Enters OSPF configuration mode.
Ruijie(config-router)#**overflow memory-lack**	Configures the OSPF system to enter the overflow state when the memory lacks.

Note By default, the OSPF system enters the overflow state automatically when the memory lacks. You can use the **no overflow memory-lack** command to disable this function.

Caution You must use the **clear ip ospf process** command or restart the OSPF protocol to exit from the overflow state.

Configuring the OSPF Network Management Function

Configuring the OSPFv2 MIB Binding

The user can only operate a sole OSPFv2 process by SNMP since the OSPFv2 MIB does not have the OSPFv2 process information. By default, the OSPFv2 MIB is bound to the OSPFv2 process with the smallest scale, and this process takes effect over all user operations.

The user can bind the OSPFv2 MIB to the process manually to operate the specified OSPFv2 process by using SNMP.

Use the following commands in routing process configuration mode.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.

Ruijie(config)# **router ospf** *1*	Enters routing protocol configuration mode.
Ruijie(config-router)# **enable mib-binding**	(Optional) Binds the OSPFv2 MIB to the specified OSPFv2 process.
Ruijie(config-router)# **end**	Returns to privileged EXEC mode.
Ruijie# **write**	Saves the configurations.

Configuring the OSPFv2 TRAP Binding

The OSPFv2 protocol defines several types of OSPF TRAP information, which is used to report various events about the OSPFv2 protocol. Sending the OSPFv2 TRAP information is not restricted by the MIB binding to the OSPFv2 process. The TRAP switch is allowed to be enabled for different processes at the same time.

Use the following commands in global configuration mode.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **snmp-server enable traps ospf**	(Optional) Enables the OSPF TRAP sending switch.
Ruijie(config)# **router ospf** *1*	Enters routing protocol configuration mode.
Ruijie(config-router)# **enable traps [error [IfAuthFailure | IfConfigError | IfRxBadPacket | VirtIfAuthFailure | VirtIfConfigError | VirtIfRxBadPacket] | lsa [LsdbApproachOverflow | LsdbOverflow | MaxAgeLsa | OriginateLsa] | retransmit [IfTxRetransmit | VirtIfTxRetransmit] | state-change [IfStateChange | NbrRestartHelperStatusChange | NbrStateChange | NssaTranslatorStatusChange | RestartStatusChange | VirtIfStateChange | VirtNbrRestartHelperStatusChange | VirtNbrStateChange]]**	(Optional) Enables the specified OSPF TRAP switch.
Ruijie(config-router)# **end**	Returns to privileged EXEC mode.
Ruijie# **write**	Saves the configurations.

Configuring the OSPF GR Function

The graceful restart (GR) function is used to enable data packets to be forwarded continuously during the restart process of the OSPF protocol. Currently, the GR function is supported on the switchover between our primary and secondary high-end devices to ensure that the key service is not interrupted.

Working Principles of the OSPF GR

OSPF GR standard:

RFC3623: Graceful OSPF Restart

Working principles of RFC3623:

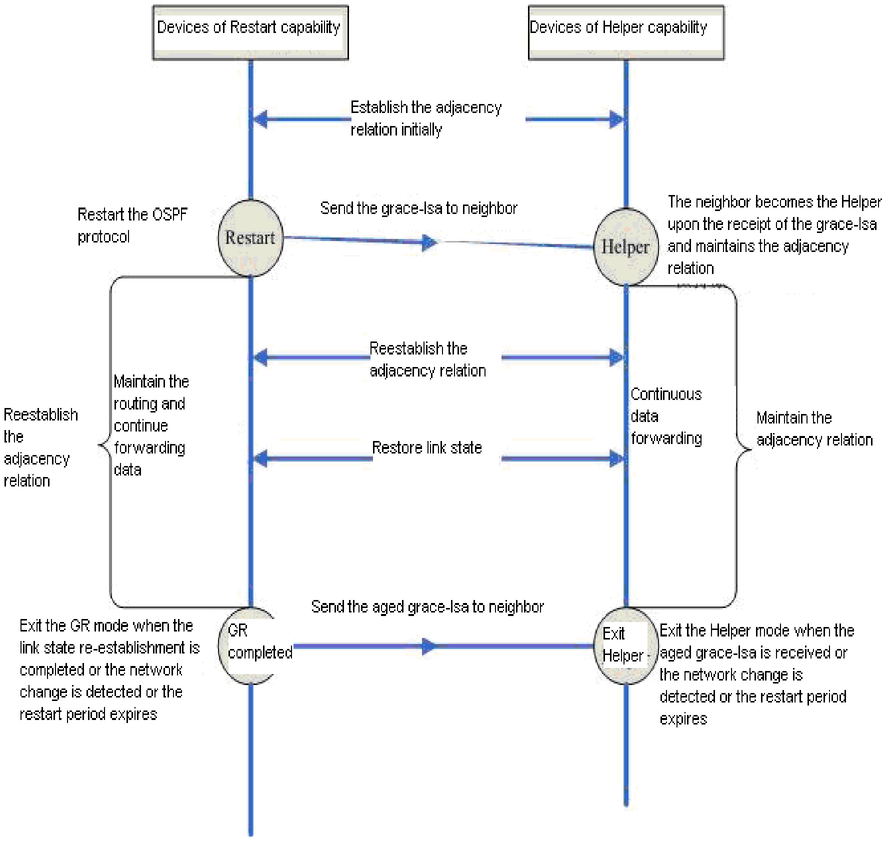
As a standard GR protocol defined by the IETF for the OSPF, RFC3623 defines the conditions, operations and precautions required for executing the Graceful Restart. As specified in RFC3623, two GR principles are important.

Namely, the network topology should be stable and the router for restarting the protocols can maintain the forwarding table during the restarting process.

The execution of OSPF GR is not an independent process. The OSPF GR has the GR Restart and GR Help functions. The device with the GR Restart capability can automatically perform the graceful restart operation, and the device with the GR Help capability can receive Grace_LSAs and help the neighbors to perform the graceful restart operation.

Generally, the device that has the GR Restart capability and is performing the GR operation is called the GR Restarter. The device that has the GR Help capability and is helping the GR Restarter to perform the GR operation is called the GR Helper. The GR process begins from the operation where the GR Restarter sends a Grace LSA. The neighbor becomes the GR Helper upon receiving the Grace LSA and assists the GR Restarter to reestablish the adjacency relationship. Meanwhile, the neighbor maintains the adjacency relationship with the GR Restarter for continuous data forwarding.

OSPF GR execution flowchart


Devices of Restart capability
Devices of Helper capability
Establish the adjacency relation initially
Restart the OSPF protocol
Restart
Send the grace-lsa to neighbor
Helper
The neighbor becomes the Helper upon the receipt of the grace-lsa and maintains the adjacency relation
Reestablish the adjacency relation
Maintain the routing and continue forwarding data
Reestablish the adjacency relation
Restore link state
Continuous data forwarding
Maintain the adjacency relation
Send the aged grace-lsa to neighbor
Exit the GR mode when the link state re-establishment is completed or the network change is detected or the restart period expires
GR completed
Exit Helper
Exit the Helper mode when the aged grace-lsa is received or the network change is detected or the restart period expires

The above figure outlines the execution process of the OSPF GR. The GR period is the longest time for reestablishing the link status. When the period for the link reestablishment or the graceful restart expires, the GR Restarter exits the GR operation.

Configuring the OSPF GR Restarter

Use the **graceful-restart** command to enable the OSPF GR restarter.

Command	Function
Ruijie # **configure terminal**	Enters global configuration mode.
Ruijie (config)# **router ospf** *1*	Enables the OSPF and enters OSPF configuration mode.

Ruijie (config-router)# **graceful-restart**	Enables the OSPF GR restarter.
Ruijie (config-router)# **end**	Returns to privileged EXEC mode.
Ruijie # **show running-config**	Verifies the configurations.
Ruijie # **write**	(Optional) Saves the configurations.

By default, the GR restarting period is 120 seconds. Use the **graceful-restart grace-period** command to modify the restarting period.

Command	Function
Ruijie # **configure terminal**	Enters global configuration mode.
Ruijie (config)# **router ospf** *1*	Enables the OSPF and enters OSPF configuration mode.
Ruijie (config-router)# **graceful-restart** grace-period *100*	Enables the OSPF GR restarter and sets the GR restarting period to 100 seconds.
Ruijie (config-router)# **end**	Returns to privileged EXEC mode.
Ruijie # **show running-config**	Verifies the configurations.
Ruijie # **write**	(Optional) Saves the configurations.

Note

The routers do not support this function.

Configuring the OSPF GR Helper

The OSPF GR Helper is enabled by default. The software provides the functions to disable the GR Helper and configure the GR Helper to detect the network changes. The following example shows how to disable and re-enable the GR Helper function and how to configure the Helper to detect the network changes.

Command	Function
Ruijie # **configure terminal**	Enters global configuration mode.
Ruijie (config)# **router ospf** *1*	Enables the OSPF and enters OSPF configuration mode.
Ruijie (config-router)# **graceful-restart helper disable**	Disables the OSPF GR Helper (for disabling the GR help to neighbors).
Ruijie (config-router)# **no graceful-restart helper disable**	Enables the OSPF GR Helper again.
Ruijie (config-router)# **graceful-restart helper** {**strict-lsa-checking** | **internal-lsa-checking**}	Enables the OSPF GR Helper to check the LSA changes to detect the network changes. If the network changes, exit the GR Helper. By default, the network changes are not detected after the GR Helper is enabled. **strict-lsa-checking:** checks the changes of types 1 to 5 and type 7 LSAs. **internal-lsa-checking:** checks the changes of types 1 to 3 LSAs.
Ruijie (config-router)# **end**	Returns to privileged EXEC mode.
Ruijie # **show running-config**	Verifies the configurations.
Ruijie # **write**	(Optional) Saves the configurations.

Changes in a part of the network may disable the GR function and cause low convergence performance on the entire network. Therefore, it is not recommended that the user enable the LSA detection function when the network is in a large scale.

Configuring the OSPF BFD Function

For details about the OSPF BFD configuration, see *BFD Configuration Guide*.

Configuring the OSPF VPN Function

For details about the OSPF VPN configuration, see *Configuring the OSPF VPN extension*.

Monitoring and Maintaining the OSPF

The following table shows the data such as the OSPF routing table, cache, and database that can be displayed.

Command	Function
show ip ospf [*process-id*]	Displays the general information about the corresponding processes of the OSPF protocol. All processes are displayed if no process number is specified.
show ip ospf [*process-id area-id*] **database** [**adv-router** *ip-address* | {**asbr-summary** | **external** | **network** | **nssa-external | opaque-area | opaque-as | opaque-link | router** | **summary**} [*link-state-id*] [{**adv-router** *ip-address* | **self-originate}**] | **database-summary** | **max-age** | **self-originate**]	Displays the OSPF database information. You can view the information about each type of LSAs in the specified process.
show ip ospf [*process-id*] **border-routers**	Shows the routing information about the specified process after reaching the ABR and ASBR.
show ip ospf interface [*interface-name*]	Shows the information about the interface involved in the OSPF routing.
show ip ospf [*process-id*] **neighbor**[*interface-name*] [*neighbor-id*] **[detail]**	Shows the information about the adjacent routers of the interface. *interface-name*: local interface connected to the neighbor *neighbor-id*: router ID of the neighbor.
show ip ospf [*process-id*] **virtual-links**	Views the virtual link information about the specified process.
show ip ospf [*process-id*] **route [count]**	Shows the routes in the OSPF routing table.
show ip ospf [*process-id*] **spf**	Shows the times for calculating inter-area routes.

For specific explanations about the commands, see *OSPF Routing Protocol Configuration Command*. The commonly used commands for monitoring and maintenance are described as follows:

5) Show the status of the OSPF neighbors.

Use the **show ip ospf** [*process-id*] **neighbor** command to show all information about neighbors in the OSPF process, including the status, role, router ID, IP address, and BFD state.

```
Ruijie# show ip ospf neighbor
```

```
OSPF process 1:
Neighbor ID     Pri State BFD State Dead Time   Address:        Interface
10.10.10.50 1 Full/DR  UP   00:00:38        10.10.10.50     eth0/0
OSPF process 100:
Neighbor ID     Pri State BFD State Dead Time   Address:    Interface
10.10.11.50 1   Full/Backup     DOWN 00:00:31   10.10.11.50     eth0/1
Ruijie# show ip ospf 1 neighbor
OSPF process 1:
Neighbor ID     Pri State BFD State Dead Time   Address:    Interface
10.10.10.50 1   Full/DR         UP 00:00:38     10.10.10.50     eth0/0
Ruijie# show ip ospf 100 neighbor
OSPF process 100:
Neighbor ID     Pri State   BFD State Dead Time     Address:    Interface
10.10.11.50 1   Full/Backup     DOWN 00:00:31   10.10.11.50     eth0/1
```

6) Show the status of the OSPF interfaces

According to the following message, the FastEthernet 0/1 interface belongs to Area 0 in the OSPF routing area, the router ID is 192.168.1.1, and the network type is BROADCAST. Pay special attention to the parameters such as *Area*, *Network Type*, *Hello*, and *Dead*. If these parameters are different from the neighbor, no adjacency relationship is established.

```
Ruijie# sh ip ospf interface fastEthernet 0/1
FastEthernet 1/0 is up, line protocol is up
Internet Address 192.168.1.1/24, Ifindex: 2 Area 0.0.0.0, MTU 1500
Matching network config: 192.168.1.0/24,
Process ID 1, Router ID 192.168.1.1, Network Type BROADCAST, Cost: 1
Transmit Delay is 1 sec, State DR, Priority 1
Designated Router (ID) 192.168.1.1, Interface Address 192.168.1.1
Backup Designated Router (ID) 192.168.1.2, Interface Address 192.168.1.2
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
Hello due in 00:00:04
Neighbor Count is 1, Adjacent neighbor count is 1
Crypt Sequence Number is 30
Hello received 972 sent 990, DD received 3 sent 4
LS-Req received 1 sent 1, LS-Upd received 10 sent 26
LS-Ack received 25 sent 7, Discarded 0
```

7) Show the information about the OSPF routing process

Run the following command to show the information about the route ID, router type, area information, and area route aggregation.

```
Ruijie# show ip ospf
Routing Process "ospf 1" with ID 1.1.1.1
Process uptime is 4 minutes
Process bound to VRF default
Conforms to RFC2328, and RFC1583 Compatibility flag is enabled
Supports only single TOS(TOS0) routes
Supports opaque LSA
```

```
Enable two-way-maintain
This router is an ASBR (injecting external routing information)
Initial SPF schedule delay 1000 msecs
Minimum hold time between two consecutive SPFs 5000 msecs
Maximum wait time between two consecutive SPFs 10000 msecs
Initial LSA throttle delay 0 msecs
Minimum hold time for LSA throttle 5000 msecs
Maximum wait time for LSA throttle 5000 msecs
Lsa Transmit Pacing timer 40 msecs, 10 LS-Upd
Minimum LSA arrival 1000 msecs
Pacing lsa-group: 240 secs
Number of incoming current DD exchange neighbors 0/5
Number of outgoing current DD exchange neighbors 0/5
Number of external LSA 4. Checksum 0x0278E0
Number of opaque AS LSA 0. Checksum 0x000000
Number of non-default external LSA 4
External LSA database is unlimited.
Number of LSA originated 6
Number of LSA received 2
Log Neighbor Adjacency Changes: Enabled
Graceful-restart disabled
Graceful-restart helper support enabled
Number of areas attached to this router: 1
Area 0 (BACKBONE)
Number of interfaces in this area is 1(1)
Number of fully adjacent neighbors in this area is 1
Area has no authentication
SPF algorithm last executed 00:01:26.640 ago
SPF algorithm executed 4 times
Number of LSA 3. Checksum 0x0204bf
Routing Process "ospf 20" with ID 2.2.2.2
Process uptime is 4 minutes
Process bound to VRF default
Conforms to RFC2328, and RFC1583 Compatibility flag is enabled
Supports only single TOS(TOS0) routes
Supports opaque LSA
Enable two-way-maintain
Initial SPF schedule delay 1000 msecs
Minimum hold time between two consecutive SPFs 5000 msecs
Maximum wait time between two consecutive SPFs 10000 msecs
Initial LSA throttle delay 0 msecs
Minimum hold time for LSA throttle 5000 msecs
Maximum wait time for LSA throttle 5000 msecs
Lsa Transmit Pacing timer 40 msecs, 10 LS-Upd
Minimum LSA arrival 1000 msecs
```

```
Pacing lsa-group: 240 secs
Number of incoming current DD exchange neighbors 0/5
Number of outgoing current DD exchange neighbors 0/5
Number of external LSA 0. Checksum 0x000000
Number of opaque AS LSA 0. Checksum 0x000000
Number of non-default external LSA 0
External LSA database is unlimited.
Number of LSA originated 0
Number of LSA received 0
Log Neighbor Adjacency Changes: Enabled
Number of areas attached to this router: 0
```

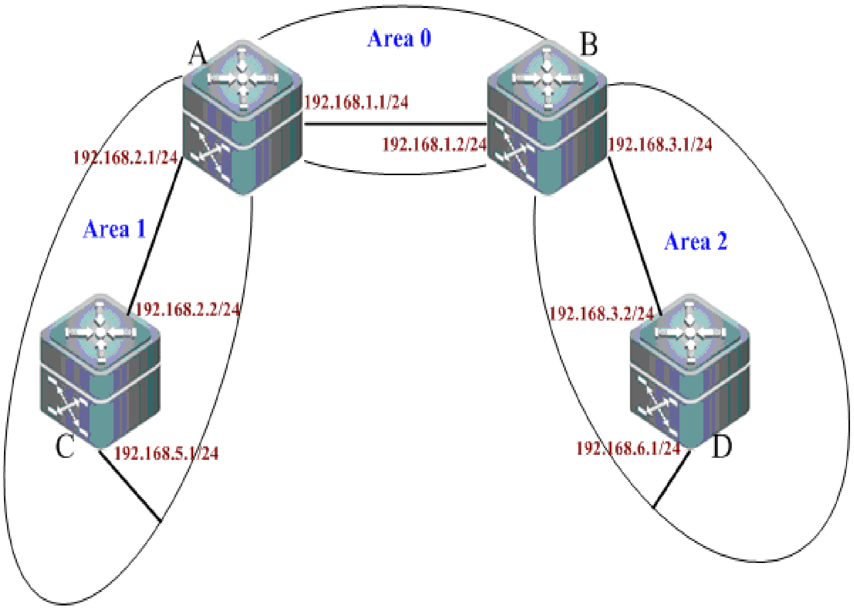
Configuration Examples

Example of Multi-Area OSPF Configuration

Networking Topology

The following figure shows the networking topology of an OSPF autonomous system. The entire autonomous system is divided into three areas: Area 0, Area 1, and Area 2. Each router runs the OSPF routing protocol.

Networking topology for the multi-area OSPF configuration



Applications

Configure Router A and Router B as area border routers (ABR) and Router C and Router D as intra-AS routers. Based on the basic OSPF configurations, every switch can successfully learn the routes in the autonomous system to all network segments.

Configuration Tips

- Configure the IP address for each interface on the routers.
- Enable the basic OSPF functions.

1. Enable the routing function (enabled by default).
2. Create an OSPF routing process.
3. Specify the IP address range associated with this routing process and the OSPF area to which the IP addresses within the range belong.

Configuration Steps

- Configuration steps of A

Step 1: Configure the IP address for the interface.

```
A(config)#interface gigabitEthernet 0/1
A(config-if-GigabitEthernet 0/1)#ip address 192.168.1.1 255.255.255.0
A(config-if-GigabitEthernet 0/1)#exit
A(config)#interface gigabitEthernet 0/2
A(config-if-GigabitEthernet 0/2)#ip address 192.168.2.1 255.255.255.0
A(config-if-GigabitEthernet 0/2)#exit
```

Step 2: Configure the basic OSPF functions.

```
A(config)#router ospf 1
A(config-router)#network 192.168.1.0 0.0.0.255 area 0
A(config-router)#network 192.168.2.0 0.0.0.255 area 1
```

- Configuration steps of B

Step 1: Configure the IP address for the interface.

```
B(config)#interface gigabitEthernet 0/1
B(config-if-GigabitEthernet 0/1)#ip address 192.168.1.2 255.255.255.0
B(config-if-GigabitEthernet 0/1)#exit
B(config)#interface gigabitEthernet 0/2
B(config-if-GigabitEthernet 0/2)#ip address 192.168.3.1 255.255.255.0
B(config-if-GigabitEthernet 0/2)#exit
```

Step 2: Configure the basic OSPF functions.

```
B(config)#router ospf 1
B(config-router)#network 192.168.1.0 0.0.0.255 area 0
B(config-router)#network 192.168.3.0 0.0.0.255 area 2
```

- Configuration steps of C

Step 1: Configure the IP address for the interface.

```
C(config)#interface gigabitEthernet 0/3
```

```
C(config-if-GigabitEthernet 0/3)#ip address 192.168.2.2 255.255.255.0
C(config-if-GigabitEthernet 0/3)#exit
C(config)#interface gigabitEthernet 0/4
C(config-if-GigabitEthernet 0/4)#ip address 192.168.5.1 255.255.255.0
C(config-if-GigabitEthernet 0/4)#exit
```

Step 2: Configure the basic OSPF functions.

```
C(config)#router ospf 1
C(config-router)#network 192.168.2.0 0.0.0.255 area 1
C(config-router)#network 192.168.5.0 0.0.0.255 area 1
```

- Configuration steps of D

Step 1: Configure the IP address for the interface.

```
D(config)#interface gigabitEthernet 0/3
D(config-if-GigabitEthernet 0/3)#ip address 192.168.3.2 255.255.255.0
D(config-if-GigabitEthernet 0/3)#exit
D(config)#interface gigabitEthernet 0/4
D(config-if-GigabitEthernet 0/4)#ip address 192.168.6.1 255.255.255.0
D(config-if-GigabitEthernet 0/4)#exit
```

Step 2: Configure the basic OSPF functions.

```
D(config)#router ospf 1
D(config-router)#network 192.168.3.0 0.0.0.255 area 2
D(config-router)#network 192.168.6.0 0.0.0.255 area 2
```

Verification

Step 1: Display information about neighbors (taking A and B as the examples).

```
A#show ip ospf neighbor
OSPF process 1, 2 Neighbors, 2 is Full:
Neighbor ID  Pri  State   Dead Time  Address        Interface
192.168.1.2 1  Full/DR  00:00:40 192.168.1.2 GigabitEthernet 0/1
192.168.2.2 1 Full/BDR  00:00:34 192.168.2.2  GigabitEthernet 0/2
B#show ip ospf neighbor
OSPF process 1, 2 Neighbors, 2 is Full:
Neighbor ID Pri  State Dead Time  Address        Interface
192.168.1.1 1 Full/BDR  00:00:32 192.168.1.1  GigabitEthernet 0/1
192.168.3.2 1 Full/BDR  00:00:30 192.168.3.2  GigabitEthernet 0/2
```

Step 2: Display OSPF routing information about A.

```
A#show ip route
Codes: C - connected, S - static, R - RIP, B - BGP
O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
```

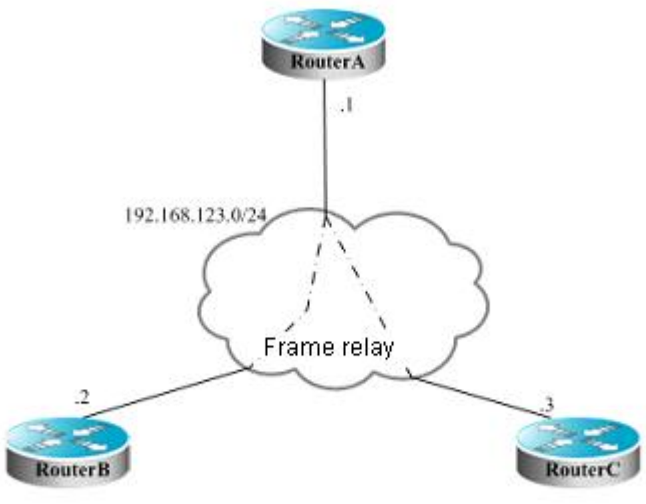
```
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default
Gateway of last resort is no set
C    192.168.1.0/24 is directly connected, GigabitEthernet 0/1
C    192.168.1.1/32 is local host.
C    192.168.2.0/24 is directly connected, GigabitEthernet 0/2
C    192.168.2.1/32 is local host.
O IA 192.168.3.0/24 [110/2] via 192.168.1.2, 00:18:03, GigabitEthernet 0/1   //inter-AS route
O    192.168.5.0/24 [110/2] via 192.168.2.2, 00:00:02, GigabitEthernet 0/2
O IA 192.168.6.0/24 [110/3] via 192.168.1.2, 00:01:02, GigabitEthernet 0/1   //inter-AS route
C#show ip route
Gateway of last resort is no set
O IA 192.168.1.0/24 [110/2] via 192.168.2.1, 00:19:05, GigabitEthernet 0/3   //inter-AS route
C    192.168.2.0/24 is directly connected, GigabitEthernet 0/3
C    192.168.2.2/32 is local host.
O IA 192.168.3.0/24 [110/3] via 192.168.2.1, 00:19:05, GigabitEthernet 0/3
C    192.168.5.0/24 is directly connected, GigabitEthernet 0/4
C    192.168.5.1/32 is local host.
O IA 192.168.6.0/24 [110/4] via 192.168.2.1, 00:03:19, GigabitEthernet 0/3   //inter-AS route
```

Example of OSPF NBMA Network Configuration

Configuration Requirements

Full mesh connection of three routers can be implemented through a frame relay network. Each router has only one frame relay link and the same link bandwidth and PVC rate. The following figure shows details about the IP address assignment and connections of the three routers.

Networking topology for OSPF NBMA network configuration



Requirements:

- The network among A, B, and C must configured as an NBMA network.

- A is the designated router, and B is the backup designated router.
- All networks are in the same area.
- Topological convergence is quickened.

Specific Configurations

Since there is no special configuration about the OSPF, you can detect neighbors in the multicast manner. If the NBMA network has been configured for the interface, the interface does not send OSPF multicast packets. Therefore, you must specify the IP addresses of neighbors. You can configure shorter SPF calculation wait-time to quicken the topological convergence.

Configurations on Router A:

# Configure the WAN interface.

```
interface Serial 1/0
ip address 192.168.123.1 255.255.255.0
encapsulation frame-relay
ip ospf network non-broadcast
ip ospf priority 10
```

# Configure the OSPF routing protocol.

```
router ospf 1
network 192.168.123.0 0.0.0.255 area 0
neighbor 192.168.123.2  priority 5
neighbor 192.168.123.3
timers throttle spf 500 1000 10000
```

Configurations on Router B:

# Configure the WAN interface.

```
interface Serial 1/0
ip address 192.168.123.2 255.255.255.0
encapsulation frame-relay
ip ospf network non-broadcast
ip ospf priority 5
```

# Configure the OSPF routing protocol.

```
router ospf 1
network 192.168.123.0 0.0.0.255 area 0
neighbor 192.168.123.1 priority 10
neighbor 192.168.123.3
timers throttle spf 500 1000 10000
```

Configurations on Router C:

# Configure the WAN interface.

```
interface Serial 1/0
```

```
ip address 192.168.123.3 255.255.255.0
encapsulation frame-relay
ip ospf network non-broadcast
```

# Configure the OSPF routing protocol.

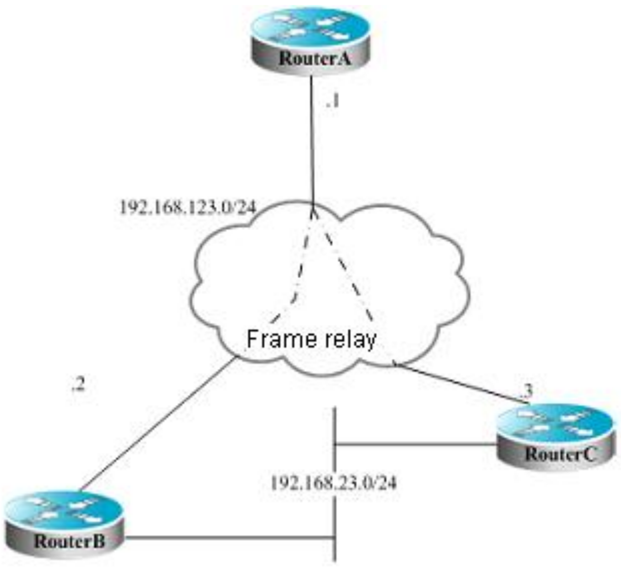
```
router ospf 1
network 192.168.123.0 0.0.0.255 area 0
neighbor 192.168.123.1 10
neighbor 192.168.123.2 5
timers throttle spf 500 1000 10000
```

Example of OSPF Point-to-multipoint Broadcasting Network Configuration

Configuration Requirements

Interconnection of three routers can be implemented through a frame relay network. Each router has only one frame relay link and the same link bandwidth and PVC rate. The following figure shows details about the IP address assignment and connections of the three routers.

Networking topology for OSPF point-to-multipoint network configuration



Requirements:

- The network among A, B, and C must configured as a point-to-multipoint network.

Specific Configurations

The point-to-multipoint network has been configured for the interface. For this network type, there is no need to specify the designated router. The OSPF operations are similar to the steps of configuring the point-to-point network.

Configurations on Router A:

# Configure the Ethernet interface.

```
interface FastEthernet 0/1
ip address 192.168.12.1 255.255.255.0
```

# Configure the WAN interface.

```
interface Serial 1/0
ip address 192.168.123.1 255.255.255.0
encapsulation frame-relay
ip ospf network point-to-multipoint
```

# Configure the OSPF routing protocol.

```
router ospf 1
network 192.168.12.0 0.0.0.255 area 0
network 192.168.123.0 0.0.0.255 area 0
```

Configurations on Router B:

# Configure the Ethernet interface.

```
interface FastEthernet 0/1
ip address 192.168.23.2 255.255.255.0
```

# Configure the WAN interface.

```
interface Serial 1/0
ip address 192.168.123.2 255.255.255.0
encapsulation frame-relay
ip ospf network point-to-multipoint
```

# Configure the OSPF routing protocol.

```
router ospf 1
network 192.168.23.0 0.0.0.255 area 0
network 192.168.123.0 0.0.0.255 area 0
```

Configurations on Router C:

# Configure the Ethernet interface.

```
interface FastEthernet 0/1
ip address 192.168.23.3 255.255.255.0
```

# Configure the WAN interface.

```
interface Serial 1/0
ip address 192.168.123.3 255.255.255.0
encapsulation frame-relay
ip ospf network point-to-multipoint
```

# Configure the OSPF routing protocol.

```
router ospf 1
network 192.168.23.0 0.0.0.255 area 0
network 192.168.123.0 0.0.0.255 area 0
```

Assuming that there is another configuration requirement for the above figure:

Router A selects Router B with priority to reach the target network 192.168.23.0/24. To meet the routing requirement, you must specify the cost for a neighbor when configuring the neighbor.

You can execute the following commands on Router A:

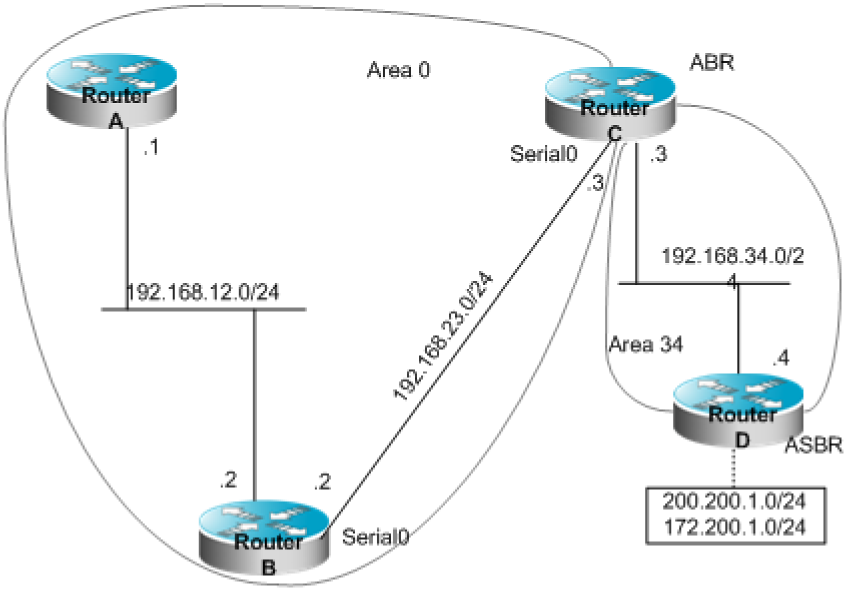
```
router ospf 1
neighbor 192.168.123.2 cost 100
neighbor 192.168.123.3 cost 200
```

Example of OSPF ABR/ASBR Configuration

Configuration Requirements

Four routers form an OSPF routing area. The networks 192.168.12.0/24 and 192.168.23.0/24 belong to Area 0, and the network 192.168.34.0/24 belongs to Area 34. The following figure shows details about the IP address assignment and router connection.

Networking topology for OSPF ABR/ASBR configuration



As shown in the figure, Router A and Router B are intra-area routers. Router C is an area border router. Router D is an AS boundary router. 200.200.1.0/24 and 172.200.1.0/24 are network segments outside the OSPF routing area. All OSPF routers shall be able to learn external routes after configuration. External routes shall be type 1 routes and carry tag 34.

Specific Configurations

While the OSPF redistributes routes of other sources, the routes to be redistributed are type-II routes and carry no tag by default.

Configurations on Router A:

# Configure the Ethernet interface.

```
interface FastEthernet 0/1
ip address 192.168.12.1 255.255.255.0
```

# Configure the OSPF routing protocol.

```
router ospf 1
network 192.168.12.0 0.0.0.255 area 0
```

Configurations on Router B:

# Configure the Ethernet interface.

```
interface FastEthernet 0/1
ip address 192.168.12.2 255.255.255.0
```

# Configure the WAN interface.

```
interface Serial 1/0
ip address 192.168.23.2 255.255.255.0
```

# Configure the OSPF routing protocol.

```
router ospf 1
network 192.168.12.0 0.0.0.255 area 0
network 192.168.23.0 0.0.0.255 area 0
```

Configurations on Router C:

# Configure the Ethernet interface.

```
interface FastEthernet 0/1
ip address 192.168.34.3 255.255.255.0
```

# Configure the WAN interface.

```
interface Serial 1/0
ip address 192.168.23.3 255.255.255.0
```

# Configure the OSPF routing protocol.

```
router ospf 1
network 192.168.23.0 0.0.0.255 area 0
network 192.168.34.0 0.0.0.255 area 34
```

Configurations on Router D:

# Configure the Ethernet interface.

```
interface FastEthernet 0/1
ip address 192.168.34.4 255.255.255.0
```

# Configure interfaces on the Ethernet adapter.

```
interface FastEthernet 0/1
ip address 200.200.1.1 255.255.255.0
interface FastEthernet 0/2
```

```
ip address 172.200.1.1 255.255.255.0
```

# Configure the OSPF routing protocol and redistribute the RIP routes.

```
router ospf 1
network 192.168.34.0 0.0.0.255 area 34
redistribute rip metric-type 1 subnets tag 34
```

# Configure the RIP routing protocol.

```
router rip
network 200.200.1.0
network 172.200.0.0
```

The OSPF routes generated on Router B are shown as follows: (Note that the type of external routes has changed to E1.)

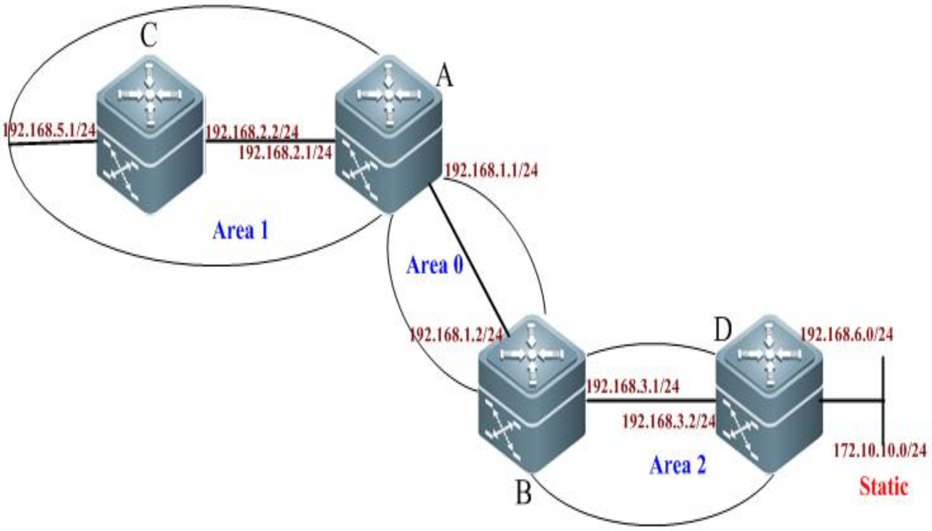
```
O E1 200.200.1.0/24 [110/85] via 192.168.23.3,00:00:33,Serial 1/0
O IA 192.168.34.0/24 [110/65] via 192.168.23.3,00:00:33,Serial 1/0
O E1 172.200.1.0 [110/85] via 192.168.23.3,00:00:33,Serial 1/0
```

Example of OSPF Static Route Redistribution Configuration

Networking Topology

The following figure shows the networking topology of an OSPF autonomous system. The entire autonomous system is divided into three areas: Area 0, Area 1, and Area 2. The network segment 172.10.10.0 is outside the routing area.

Networking topology for OSPF static route redistribution configuration



Applications

Configure Router A and Router B as area border routers (ABR) and Router C as an intra-area router. Configure Router D as an ASBR and introduce an external static route, so that all OSPF routers in non-stub area can successfully learn this external route.

Configuration Tips

- Configure the IP address for interfaces on the routers (omitted).
- Configure the basic OSPF functions (see "Example of Multi-Area OSPF Configuration")
- Introduce and configure the external static route.

Configuration Steps

Step 1: On Router D, configure a static route to the network segment 172.10.10.0.

```
D(config)#ip route 172.10.10.0 255.255.255.0 192.168.6.2
```

Step 2: Display the routing table of Router A.

```
A#show ip route ospf
O IA 192.168.3.0/24 [110/2] via 192.168.1.2, 15:33:00, GigabitEthernet 0/1
O    192.168.5.0/24 [110/2] via 192.168.2.2, 15:14:59, GigabitEthernet 0/2
O IA 192.168.6.0/24 [110/3] via 192.168.1.2, 00:17:58, GigabitEthernet 0/1
```

In this case, there is no route to the network segment 172.10.10.0.

Step 3: Redistribute the static route on Router D

```
D(config)#router ospf 1
D(config-router)# redistribute static subnets
```

Verification

Step 1: Display the routing table of Router D.

```
D#show ip route
Codes:  C - connected, S - static, R - RIP, B - BGP
O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
S    172.10.10.0/24 [1/0] via 192.168.6.2
O IA 192.168.1.0/24 [110/2] via 192.168.3.1, 15:25:19, GigabitEthernet 0/3
O IA 192.168.2.0/24 [110/3] via 192.168.3.1, 15:25:19, GigabitEthernet 0/3
C    192.168.3.0/24 is directly connected, GigabitEthernet 0/3
C    192.168.3.2/32 is local host.
O IA 192.168.5.0/24 [110/4] via 192.168.3.1, 15:11:56, GigabitEthernet 0/3
C    192.168.6.0/24 is directly connected, GigabitEthernet 0/4
C    192.168.6.1/32 is local host.
```

Step 2: View OSPF information about Router D. Key point: Router D is an AS boundary router (ASBR).

```
D#show ip ospf
```

```
 Routing Process "ospf 1" with ID 192.168.3.2
 Process uptime is 15 hours 27 minutes
 Process bound to VRF default
 Conforms to RFC2328, and RFC1583 Compatibility flag is enabled
 Supports only single TOS(TOS0) routes
 Supports opaque LSA
 Enable two-way-maintain
 This router is an ASBR (injecting external routing information)
 Initial SPF schedule delay 1000 msecs
Minimum hold time between two consecutive SPFs 5000 msecs
Maximum wait time between two consecutive SPFs 10000 msecs
Initial LSA throttle delay 0 msecs
Minimum hold time for LSA throttle 5000 msecs
Maximum wait time for LSA throttle 5000 msecs
Lsa Transmit Pacing timer 40 msecs, 10 LS-Upd
Minimum LSA arrival 1000 msecs
 Pacing lsa-group: 240 secs
 Number of incoming current DD exchange neighbors 0/5
 Number of outgoing current DD exchange neighbors 0/5
 Number of external LSA 1. Checksum 0x006DB0
 Number of opaque AS LSA 0. Checksum 0x000000
 Number of non-default external LSA 1
 External LSA database is unlimited.
 Number of LSA originated 2
 Number of LSA received 173
 Log Neighbor Adjacency Changes: Enabled
  Number of areas attached to this router: 1: 1 normal 0 stub 0 nssa
    Area 2
        Number of interfaces in this area is 2(2)
        Number of fully adjacent neighbors in this area is 1
        Number of fully adjacent virtual neighbors through this area is 0
        Area has no authentication
        SPF algorithm last executed 00:06:27.540 ago
        SPF algorithm executed 9 times
        Number of LSA 6. Checksum 0x0212ff
```

Step 3: Display the routing table of Router A.

```
A#show ip route ospf
O E2 172.10.10.0/24 [110/20] via 192.168.1.2, 00:07:37, GigabitEthernet 0/1
O IA 192.168.3.0/24 [110/2] via 192.168.1.2, 15:33:00, GigabitEthernet 0/1
O    192.168.5.0/24 [110/2] via 192.168.2.2, 15:14:59, GigabitEthernet 0/2
O IA 192.168.6.0/24 [110/3] via 192.168.1.2, 00:17:58, GigabitEthernet 0/1
```

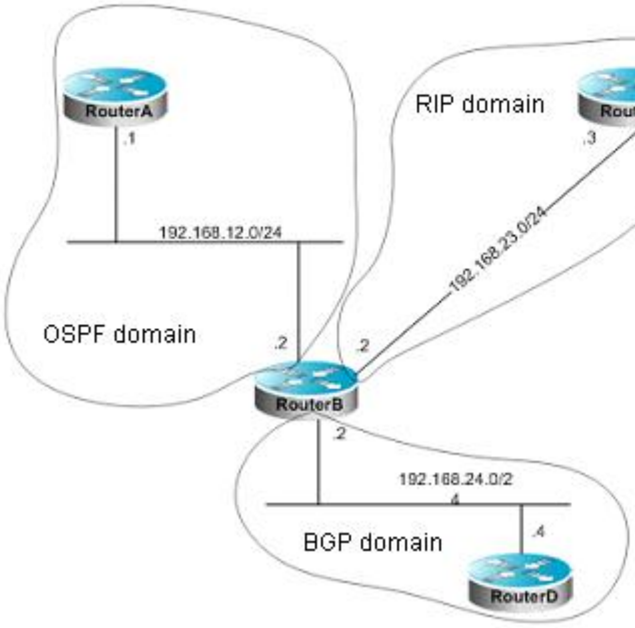
In this case, Router A has successfully learned the route to the network segment 172.10.10.0.

Example of OSPF Dynamic Route Redistribution Configuration

Configuration Requirements

The following figure shows the topology of four routers. Router A belongs to the OSPF routing area. Router C belongs to the RIP routing area. Router D belongs to the BGP routing area. Router B is connected to the three routing areas. Router A advertises two routes: 192.168.10.0/24 and 192.168.100.1/32. Router C advertises two routes: 192.168.3.0/24 and 192.168.30.0/24. Router D advertises two routes: 192.168.4.0/24 and 192.168.40.0/24.

Networking topology for dynamic routing protocol redistribution



On Router B, the OSPF redistributes routes (type-1) in the RIP routing area and the BGP routes that carry the community attribute 11:11 in the BGP routing area. The RIP redistributes the route 192.168.10.0/24 in the OSPF routing area and advertises a default route to the RIP routing area. The metric of this route is set to 2.

Specific Configurations

When the routing protocol redistribute the routes among each other, the simple route filtering can be controlled by using the distribution list. However, different attributes must be set for different routes, which cannot be implemented by using the distribution list. In this case, a route-map must be used for control. The route-map provides more control functions than the distribution list, but the router configuration is more complex. Therefore, do not use the route-map if possible. The following examples use the route-map to match the community attribute of the BGP routes.

Configurations on Router A:

# Configure the network interface.

```
Ruijie(config)# interface gigabitEthernet 0/1
Ruijie(config-if-GigabitEthernet 0/1)# ip address 192.168.10.1 255.255.255.0
Ruijie(config)# interface loopback 1
Ruijie(config-if-Loopback 1)# ip address 192.168.100.1 255.255.255.255
Ruijie(config-if-Loopback 1)# no ip directed-broadcast
Ruijie(config)# interface gigabitEthernet 0/2
Ruijie(config-if-GigabitEthernet 0/2)# ip address 192.168.12.1 255.255.255.0
```

# Configure the OSPF.

```
Ruijie(config)# router ospf 12
Ruijie(config-router)# network 192.168.10.0 0.0.0.255 area 0
Ruijie(config-router)# network 192.168.12.0 0.0.0.255 area 0
Ruijie(config-router)# network 192.168.100.0 0.0.0.255 area 0
```

Configurations on Router B:

# Configure the network interface.

```
Ruijie(config)# interface gigabitEthernet 0/1
Ruijie(config-if-GigabitEthernet 0/1)# ip address 192.168.12.2 255.255.255.0
Ruijie(config)# interface gigabitEthernet 0/2
Ruijie(config-if-GigabitEthernet 0/2)# ip address 192.168.24.2
Ruijie(config)# interface Serial 1/0
Ruijie(config-Serial 1/0)# ip address 192.168.23.2 255.255.255.0
```

# Configure the OSPF and specify the type of routes to be redistributed.

```
Ruijie(config)# router ospf 12
Ruijie(config-router)# redistribute rip metric 100 metric-type 1 subnets
Ruijie(config-router)# redistribute bgp route-map ospfrm subnets
Ruijie(config-router)# network 192.168.12.0 0.0.0.255 area 0
```

# Configure the RIP and use the distribute list to filter the redistributed routes.

```
Ruijie(config)# router rip
Ruijie(config-router)# redistribute ospf 12 metric 2
Ruijie(config-router)# network 192.168.23.0
Ruijie(config-router)# distribute-list 10 out ospf
Ruijie(config-router)# default-information originate always
Ruijie(config-router)# no auto-summary
```

# Configure the BGP.

```
Ruijie(config)# router bgp 2
Ruijie(config-router)# neighbor 192.168.24.4 remote-as 4
Ruijie(config-router)# address-family ipv4
Ruijie(config-router-af)# neighbor 192.168.24.4 activate
Ruijie(config-router-af)# neighbor 192.168.24.4  send-community
```

# Configure the route-map.

```
Ruijie(config)# route-map ospfrm
Ruijie(config-route-map)# match community cl_110
```

# Define the access list.

```
Ruijie(config)# access-list 10 permit 192.168.10.0
```

# Define the community list.

```
Ruijie(config)# ip community-list standard cl_110 permit 11:11
```

Configurations on Router C:

# Configure the network interface.

```
Ruijie(config)# interface gigabitEthernet 0/1
Ruijie(config-if-GigabitEthernet 0/1)# ip address 192.168.30.1 255.255.255.0
Ruijie(config)# interface gigabitEthernet 0/2
Ruijie(config-if-GigabitEthernet 0/2)# ip address 192.168.3.1 255.255.255.0
Ruijie(config)# interface Serial 1/0
Ruijie(config-if-Serial 1/0)# ip address 192.168.23.3 255.255.255.0
```

# Configure the RIP.

```
Ruijie(config)# router rip
Ruijie(config-router)# network 192.168.23.0
Ruijie(config-router)# network 192.168.3.0
Ruijie(config-router)# network 192.168.30.0
```

Configurations on Router D:

# Configure the network interface.

```
Ruijie(config)# interface gigabitEthernet 0/1
Ruijie(config-if-GigabitEthernet 0/1)# ip address 192.168.40.1 255.255.255.0
Ruijie(config)# interface gigabitEthernet 0/2
Ruijie(config-if-GigabitEthernet 0/2)# ip address 192.168.4.1 255.255.255.0
Ruijie(config)# interface gigabitEthernet 0/3
Ruijie(config-if-GigabitEthernet 0/3)# ip address 192.168.24.4 255.255.255.0
```

# Configure the BGP.

```
Ruijie(config)# router bgp 4
Ruijie(config-router)# neighbor 192.168.24.2 remote-as 2
Ruijie(config-router)# redistribute connected route-map bgprm
Ruijie(config-router)# address-family ipv4
Ruijie(config-router-af)# neighbor 192.168.24.2 activate
Ruijie(config-router-af)# neighbor 192.168.24.2 send-community
```

# Configure the route-map.

```
Ruijie(config)# route-map bgprm
Ruijie(config-route-map)# set community 22:22
```

The OSPF routes learned by Router A:

```
O E1 192.168.30.0/24[110/101]via 192.168.12.2,00:04:07, gigabitEthernet 0/2
O E1 192.168.3.0/24[110/101]via 192.168.12.2,00:04:07, gigabitEthernet 0/2
O E1 192.168.23.0/24[110/101]via 192.168.12.2,00:04:07, gigabitEthernet 0/2
```

The RIP routes learned by Router C:

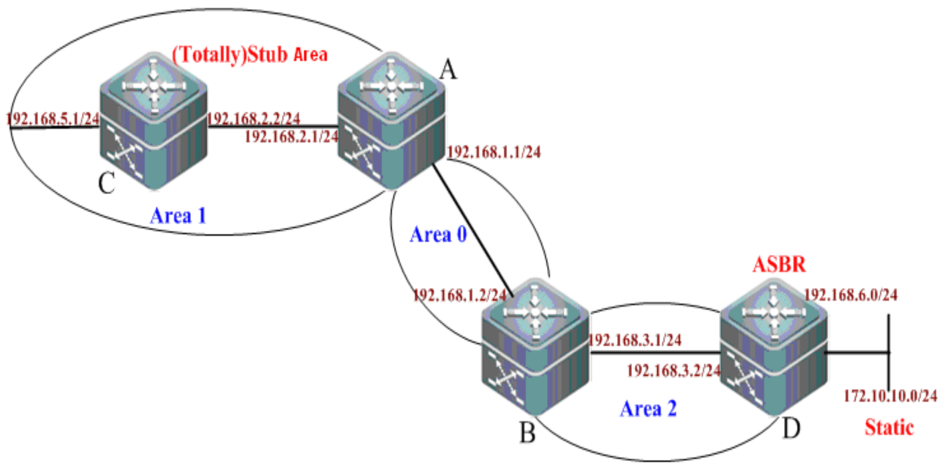
```
R    0.0.0.0/0 [120/1] via 192.168.23.2, 00:00:00, Serial 1/0
R    192.168.10.0/24 [120/2] via 192.168.23.2, 00:00:00, Serial 1/0
```

Example of OSPF (Totally) Stub Area Configuration

Networking Topology

The following figure shows the networking topology of an OSPF autonomous system. The entire autonomous system is divided into three areas: Area 0, Area 1, and Area 2. The network segment 172.10.10.0 is outside the routing area.

Networking topology for OSPF (Totally) Stub area configuration



Applications

Configure Router A and Router B as area border routers (ABR) and Router C as an intra-area router. Configure Router D as an ASBR and introduce one an external static route.

To reduce the size of the routing table inside the AS border and the number of routes exchanged, configure the specific area to be a (Totally) Stub area.

Routing information can be correctly transmitted in the OSPF autonomous system.

Configuration Tips

Do not configure the backbone area (Area 0) cannot be configured as a (Totally) Stub area, and there must be no ASBR exists in the (Totally) Stub area. That is, the external routes of the autonomous system cannot be propagated transmitted in this area. In this example, Area 1 is configured as the (Totally) Stub area.

When configuring an area as a Stub area, you must configure the **stub** command on all routers in this area. In this example, you need to configure this attribute on Router A and Router C.

When configuring an area as a Totally Stub area, you must configure the **stub** command on all routers (Router C) in this area and the **stub [no-summary]** command on the ABR (Router A).

Configuration Steps

The following information only shows how to configure an OSPF (Totally) Stub area. For other configurations, see "Example of Multi-Area OSPF Configuration" and "Example of OSPF Static Route Redistribution Configuration".

Step 1: Display the routing table of Router C when this router is in a normal area.

```
C#show ip route  ospf
O E2 172.10.10.0/24 [110/20] via 192.168.2.1, 4d,02:28:07, GigabitEthernet 0/3
                        //AS external route
O IA 192.168.1.0/24 [110/2] via 192.168.2.1, 4d,17:52:14, GigabitEthernet 0/3
O IA 192.168.3.0/24 [110/3] via 192.168.2.1, 4d,17:52:14, GigabitEthernet 0/3
O IA 192.168.6.0/24 [110/4] via 192.168.2.1, 4d,02:38:27, GigabitEthernet 0/3
```

In this case, the routing table contains AS external routes.

Step 2: Configure the Stub area

- Configurations on Router A:

```
A(config)#router ospf 1
A(config-router)#area 1 stub
```

- Configurations on Router C:

```
C(config)#router ospf 1
C(config-router)#area 1 stub
```

- Display the routing table of Router C when this router is in a stub area.

```
C#show ip route ospf
O*IA 0.0.0.0/0 [110/2] via 192.168.2.1, 00:00:32, GigabitEthernet 0/3
                                        //default route
O IA 192.168.1.0/24 [110/2] via 192.168.2.1, 00:00:32, GigabitEthernet 0/3
O IA 192.168.3.0/24 [110/3] via 192.168.2.1, 00:00:32, GigabitEthernet 0/3
O IA 192.168.6.0/24 [110/4] via 192.168.2.1, 00:00:32, GigabitEthernet 0/3
```

In this case, the routing table contains no AS external route. The original AS external route in the routing table has been replaced with a default route.

Step 3: Configure the Totally Stub area

- Configurations on Router A:

```
A(config)#router ospf 1
A(config-router)#area 1 stub stub no-summary
```

- Configurations on Router C:

```
C(config)#router ospf 1
C(config-router)#area 1 stub
```

- Display the routing table of Router C when this router is in a totally stub area.

```
C#show ip route ospf
```

```
O*IA 0.0.0.0/0 [110/2] via 192.168.2.1, 00:30:53, GigabitEthernet 0/3
```

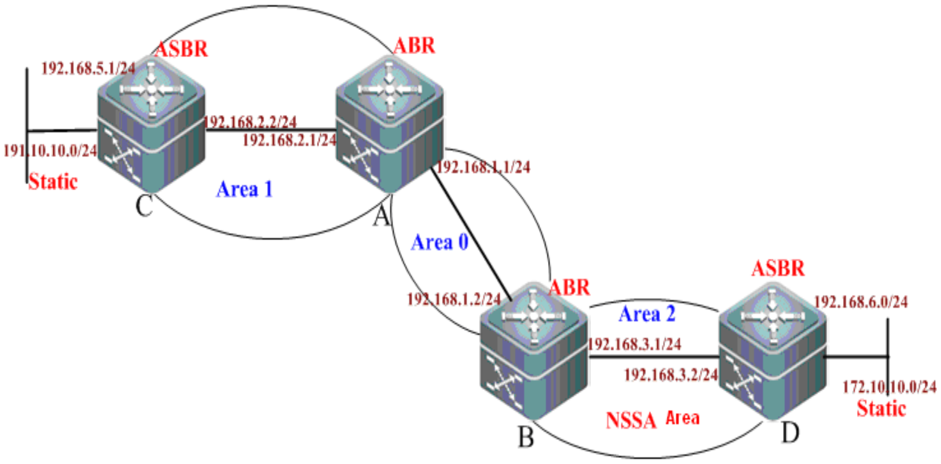
In this case, the routing table only contains one default route to the external area.

Example of OSPF NSSA Area Configuration

Networking topology

The following figure shows the networking topology of an OSPF autonomous system. The entire autonomous system is divided into three areas: Area 0, Area 1, and Area 2. The network segments 192.10.10.0 and 172.10.10.0 are outside the OSPF routing area.

Networking topology for OSPF NSSA area configuration



Applications

1. Configure Router A and Router B as area border routers (ABR). Configure Router C and Router D as ASBRs and introduce an AS external static route for Router C and Router D respectively.

2. Area 2 shall be configured as an NSSA area in order to reduce the size of routing table of the intra-area router and the number of routes exchanged. Meanwhile, prohibit Router B from sending summary LSAs (Type-3 LSA) to the NSSA area.

3. Routing information can be correctly transmitted in the OSPF autonomous system.

Configuration Tips

Tips for configuring the NSSA area are as follows:

1. The backbone area (Area 0) cannot be configured as the NSSA area.

2. The ASBRs can exist in the NSSA area, and certain number of AS external routes can be imported to the OSPF routing area.

3. When configuring an area as the NSSA area, you must use the area nssa command on all routers (Router B and D) connected to the NSSA area.

Configuration Steps

The following information only shows how to configure the NSSA area. For the basic OSPF configurations, see the above examples.

Step 1: Configure static route redistribution.

- Configurations on Router C:

! Configure a static route.

```
C(config)#ip route 191.10.10.0 255.255.255.0 192.168.5.2
```

! Redistribute the static route based on the OSPF.

```
C(config)#router ospf 1
C(config-router)#redistribute static subnets
```

- Configurations on Router D:

! Configure a static route.

```
D(config)#ip route 172.10.10.0 255.255.255.0 192.168.6.2
```

! Redistribute the static route based on the OSPF.

```
C(config)#router ospf 1
C(config-router)#redistribute static subnets
```

Step 2: Configure the NSSA.

- Configurations on Router B (ABR):

```
B(config)#router ospf 1
```

! Define the NSSA area and prohibit this ABR from sending summary LSAs (Type-3 LSA) to the NSSA area.

```
B(config-router)#area 2 nssa no-summary
```

- Configurations on Router D (ASBR):

```
D(config)#router ospf 1
D(config-router)#area 2 nssa
```

Verification

Step 1: Display the routing information when Area 2 is configured as a normal area.

- Display the routing table of Router D (ASBR).

```
D#show ip route
Codes:  C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default
Gateway of last resort is no set
S    172.10.10.0/24 [1/0] via 192.168.6.2
O E2 191.10.10.0/24 [110/20] via 192.168.3.1, 00:00:21, GigabitEthernet 0/3
O IA 192.168.1.0/24 [110/2] via 192.168.3.1, 00:00:21, GigabitEthernet 0/3
O IA 192.168.2.0/24 [110/3] via 192.168.3.1, 00:00:21, GigabitEthernet 0/3
C    192.168.3.0/24 is directly connected, GigabitEthernet 0/3
C    192.168.3.2/32 is local host.
O IA 192.168.5.0/24 [110/4] via 192.168.3.1, 00:00:21, GigabitEthernet 0/3
C    192.168.6.0/24 is directly connected, GigabitEthernet 0/4
C    192.168.6.1/32 is local host.
```

- Display the OSPF routing table of Router B (ABR).

```
B#show ip route ospf
O E2 172.10.10.0/24 [110/20] via 192.168.3.2, 17:53:35, GigabitEthernet 0/2    O E2
191.10.10.0/24 [110/20] via 192.168.1.1, 00:57:46, GigabitEthernet 0/1
O IA 192.168.2.0/24 [110/2] via 192.168.1.1, 5d,15:39:01, GigabitEthernet 0/1
O IA 192.168.5.0/24 [110/3] via 192.168.1.1, 01:10:34, GigabitEthernet 0/1
O    192.168.6.0/24 [110/2] via 192.168.3.2, 17:53:36, GigabitEthernet 0/2
```

Step 2: Display the routing information about each router in the NSSA area when Area 2 is configured as an NSSA area.

- Display the OSPF routing table of Router B (ABR).

```
B#show ip route ospf
O N2 172.10.10.0/24 [110/20] via 192.168.3.2, 00:01:00, GigabitEthernet 0/2
O E2 191.10.10.0/24 [110/20] via 192.168.1.1, 01:11:26, GigabitEthernet 0/1
O IA 192.168.2.0/24 [110/2] via 192.168.1.1, 5d,15:52:41, GigabitEthernet 0/1
O IA 192.168.5.0/24 [110/3] via 192.168.1.1, 01:24:14, GigabitEthernet 0/1
O    192.168.6.0/24 [110/2] via 192.168.3.2, 00:01:01, GigabitEthernet 0/2
```

In this case, the ABR in the NSSA area has translated the AS external routes imported into this area into N2 (OSPF NSSA external type 2) routes and transmitted to other areas.

- Display the routing table of Router D (ASBR).

```
D#show ip route
Codes:  C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default
Gateway of last resort is no set
S    172.10.10.0/24 [1/0] via 192.168.6.2
O IA 192.168.1.0/24 [110/2] via 192.168.3.1, 00:03:20, GigabitEthernet 0/3
O IA 192.168.2.0/24 [110/3] via 192.168.3.1, 00:03:20, GigabitEthernet 0/3
C    192.168.3.0/24 is directly connected, GigabitEthernet 0/3
```

```
C   192.168.3.2/32 is local host.
O IA 192.168.5.0/24 [110/4] via 192.168.3.1, 00:03:20, GigabitEthernet 0/3
C   192.168.6.0/24 is directly connected, GigabitEthernet 0/4
C   192.168.6.1/32 is local host.
```

In this case, the AS external routes imported into other areas cannot reach this area when Router D is in an NSSA area.

Step 3: Display the routing information of the NSSA area when configuring the attribute of the NSSA area to be no-summary on Router B (ABR).

```
D#show ip route
Codes:  C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default
Gateway of last resort is 192.168.3.1 to network 0.0.0.0
O*IA 0.0.0.0/0 [110/2] via 192.168.3.1, 00:00:40, GigabitEthernet 0/3
S   172.10.10.0/24 [1/0] via 192.168.6.2
C   192.168.3.0/24 is directly connected, GigabitEthernet 0/3
C   192.168.3.2/32 is local host.
C   192.168.6.0/24 is directly connected, GigabitEthernet 0/4
C   192.168.6.1/32 is local host.
```

In this case, the routing table contains a default route which replaces the inter-area route.

Step 4: Display the OSPF routing information on routers in other areas. Key point: Note whether there is any AS external route imported into the NSSA area.

```
SwitchA#show ip route ospf
O E2 172.10.10.0/24 [110/20] via 192.168.1.2, 02:08:08, GigabitEthernet 0/1
O E2 191.10.10.0/24 [110/20] via 192.168.2.2, 03:18:35, GigabitEthernet 0/2
O IA 192.168.3.0/24 [110/2] via 192.168.1.2, 5d,17:59:01, GigabitEthernet 0/1
O   192.168.5.0/24 [110/2] via 192.168.2.2, 03:31:25, GigabitEthernet 0/2
O IA 192.168.6.0/24 [110/3] via 192.168.1.2, 02:08:09, GigabitEthernet 0/1
```

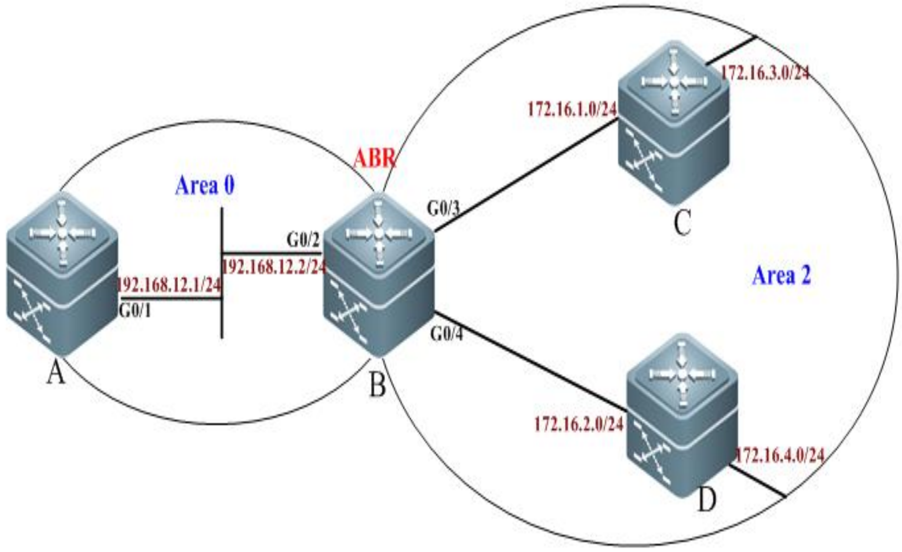
In this case, the routing table of Router A contains an AS external route imported into the NSSA area.

Example of OSPF Inter-area Route Aggregation Configuration

Networking Topology

The following figure shows the topological topology of an OSPF autonomous system, in which the network segment 192.168.12.0/24 belongs to Area 0 and the network segments 172.16.1.0/24, 172.16.2.0/24, 172.16.3.0/24, and 172.16.4.0/24 belong to Area 2.

Networking topology for OSPF inter-area route aggregation configuration



Applications

To reduce the size of routing table, configure Router B so that Router B only advertises the summary route of four network segments (172.16.1.0/24, 172.16.2.0/24, 172.16.3.0/24, and 172.16.4.0/24) instead of separately advertising the routes of these four network segments.

Configuration Tips

1. Since the network segments 172.16.1.0/24, 172.16.2.0/24, 172.16.3.0/24, and 172.16.4.0/24 are consecutive addresses, you can configure route aggregation on the area border router (Router B) to alleviate route calculation. Use this command **area range** to configure the route aggregation between the OSPF inter-areas".

2. During route aggregation, the aggregated route range may exceed the actual network range in the routing table. Routing loop may incur or load on the router may increase if packets are sent to a network that is beyond the aggregated route range. So you need to add a "discard" route into the routing table on the ABR (Router B) or ASBR. Use the inter-area route aggregation command area range to add the discard route. This function is enabled by default.

3. The aggregated route address of 172.16.1.0/24, 172.16.2.0/24, 172.16.3.0/24, and 172.16.4.0/24 is 172.16.0.0/21. Routes falling within this range will not be advertised to other areas by the ABR.

Configuration Steps

Step 1: Configure the IP address for interfaces.(omitted)

Step 2: Configure the basic OSPF functions.

- Configure Router A.

```
A(config)#router ospf 1
A(config)# network 192.168.12.0 0.0.0.255 area 0
```

- Configure Router B.

```
B(config)#router ospf 1
```

```
B(config-router)#network 192.168.12.0 0.0.0.255 area 0
B(config-router)#network 172.16.1.0 0.0.0.255 area 2
B(config-router)#network 172.16.2.0 0.0.0.255 area 2
```

- Configure Router C.

```
C(config)#router ospf 1
C(config-router)#network 172.16.1.0 0.0.0.255 area 2
C(config-router)#network 172.16.3.0 0.0.0.255 area 2
```

- Configure Router D.

```
D(config)#router ospf 1
D(config-router)#network 172.16.2.0 0.0.0.255 area 2
D(config-router)#network 172.16.4.0 0.0.0.255 area 2
```

- Display the OSPF routing table of Router A.

```
A#show ip route ospf
O IA 172.16.1.0/24 [110/2] via 192.168.12.2, 00:06:47, GigabitEthernet 0/1
O IA 172.16.2.0/24 [110/2] via 192.168.12.2, 00:06:47, GigabitEthernet 0/1
O IA 172.16.3.0/24 [110/3] via 192.168.12.2, 00:06:47, GigabitEthernet 0/1
O IA 172.16.4.0/24 [110/3] via 192.168.12.2, 00:06:19, GigabitEthernet 0/1
```

In this case, the detailed routing information about Area 2 is advertised to Area 0.

Step 3: Configure the inter-area route aggregation on the ABR (Router B).

```
B(config)#router ospf 1
B(config-router)#area 2 range 172.16.0.0 255.255.248.0
```

Step 4: On ABR (Router B), configure to control the addition of the aggregated route entry into the core routing table. This function is enabled by default.

```
B(config-router)# discard-route internal
```

Verification

- After configuring route aggregation, display the OSPF routing table of Router A.

```
A#show ip route ospf
O IA 172.16.0.0/21 [110/2] via 192.168.12.2, 00:01:04, GigabitEthernet 0/1
```

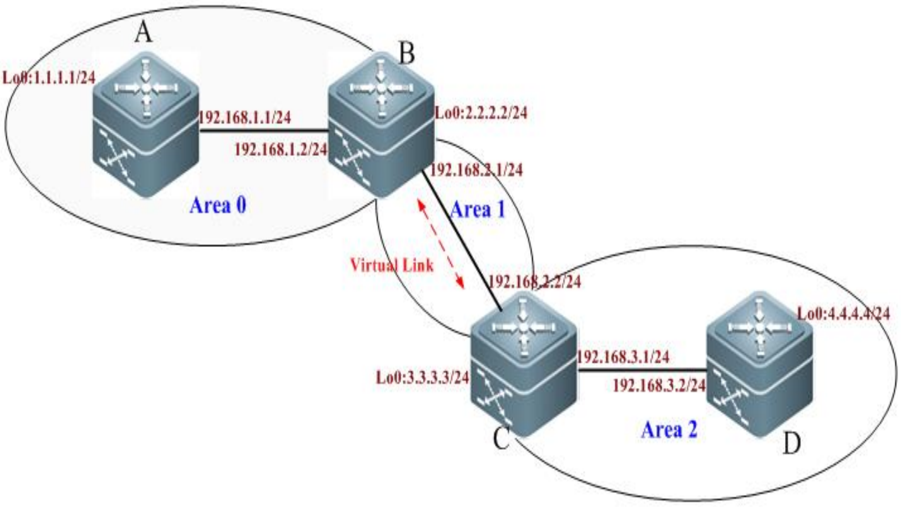
In this case, only the aggregated routes are advertised. Specific routes will not be advertised by the ABR to other areas. The size of the routing table is decreased substantially.

Example of OSPF Virtual Link Configuration

Networking Topology

The following figure shows an OSPF routing area. The network segment 192.168.1.0 belongs to Area 0. The network segment 192.168.2.0 belongs to Area 1. The network segment 192.168.3.0 belongs to Area 2. Due to the limitation of physical conditions, other specific areas cannot be deployed around the backbone area. As shown in the following figure, Area 2 is not directly connected to Area 0.

Networking topology for OSPF virtual link configuration



Applications

Through configuration, Router D shall be able to receive routes of the network segments 192.168.1.0/24 (Area 0) and 192.168.2.0/24 (Area 1). Meanwhile, Router B shall be able to learn the routes of the network segment 192.168.3.0/24 (Area 2).

Details about IP address assignment are shown as follows:

Router name	Router ID	Interface address
A	1.1.1.1	Gi0/1: 192.168.1.1/24
B	2.2.2.2	Gi0/1: 192.168.1.2/24 Gi0/3: 192.168.2.1/24
C	3.3.3.3	Gi0/3: 192.168.2.2/24 Gi0/5: 192.168.3.1/24
D	4.4.4.4	Gi0/5: 192.168.3.2/24

Configuration Tips

When the OSPF routing area is composed of multiple areas, each area must be directly connected to the backbone area (Area 0). Otherwise, these areas cannot be interconnected. If there is no direct physical link, create virtual links to logically connect each area to the backbone area. Configuration tips are shown as follows:

- Configure the IP address for the interfaces. (Omitted)
- Configure the basic OSPF functions.
- Configure OSPF virtual links

The virtual link must be configured on ABRs. This example configures virtual links on Router B and Router C.

Use the **area** *area-id* **virtual-link** *router-id* command to configure virtual links on the ABRs. The router-id refers to the identifier of a peer device.

Configuration Steps

Step 1: Configure the basic OSPF functions.

➢ Configurations on Router A:

! Create an OSPF routing process and specify the IP address range associated with this routing process and the OSPF area to which these IP addresses belong.

```
A(config)#router ospf 1
A(config-router)#network 192.168.1.0 0.0.0.255 area 0
A(config-router)#exit
```

! Configure the loopback IP 1.1.1.1 as the router ID of Router A.

```
A(config)#interface loopback 0
A(config-Loopback 0)#ip address 1.1.1.1 255.255.255.0
```

➢ Configurations on Router B:

! Create an OSPF routing process and specify the IP address range associated with this routing process and the OSPF area to which these IP addresses belong.

```
B(config)#router ospf 1
B(config-router)#network 192.168.1.0 0.0.0.255 area 0
B(config-router)#network 192.168.2.0 0.0.0.255 area 1
B(config-router)#exit
```

! Configure the loopback IP 2.2.2.2 as the router ID of Router B.

```
B(config)#interface loopback 0
B(config-Loopback 0)#ip address 2.2.2.2 255.255.255.0
```

➢ Configurations on Router C:

! Create an OSPF routing process and specify the IP address range associated with this routing process and the OSPF area to which these IP addresses belong.

```
C(config)#router ospf 1
C(config-router)#network 192.168.2.0 0.0.0.255 area 1
C(config-router)#network 192.168.3.0 0.0.0.255 area 2
C(config-router)#exit
```

! Configure the loopback IP 3.3.3.3 as the router ID of Router C.

```
C(config)#interface loopback 0
C(config-Loopback 0)#ip address 3.3.3.3 255.255.255.0
```

➢ Configurations on Router D:

! Create an OSPF routing process and specify the IP address range associated with this routing process and the OSPF area to which these IP addresses belong.

```
D(config)#router ospf 1
D(config-router)#network 192.168.3.0 0.0.0.255 area 2
```

```
D(config-router)#exit
```

! Configure the loopback IP 4.4.4.4 as the router ID of Router D.

```
D(config)#interface loopback 0
D(config-Loopback 0)#ip address 4.4.4.4 255.255.255.0
```

- Display the OSPF routing table of Router A.

```
A#show ip route ospf
O IA 192.168.2.0/24 [110/2] via 192.168.1.2, 00:32:48, GigabitEthernet 0/1
```

Since Area 2 is not directly connected to Area 0, the routing table of Router A contains no routing information about Area 2

Step 2: Configure OSPF virtual links.

- Configure Router B.

```
B(config)#router ospf 1
B(config-router)#area 1 virtual-link 3.3.3.3
```

- Configure Router C.

```
C(config)#router ospf 1
C(config-router)#area 1 virtual-link 2.2.2.2
```

Verification

- Display the OSPF routing table of Router B.

```
B#show ip route
Codes:  C - connected, S - static, R - RIP, B - BGP
O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default
Gateway of last resort is no set
C    2.2.2.0/24 is directly connected, Loopback 0
C    2.2.2.2/32 is local host.
C    192.168.1.0/24 is directly connected, GigabitEthernet 0/1
C    192.168.1.2/32 is local host.
C    192.168.2.0/24 is directly connected, GigabitEthernet 0/3
C    192.168.2.1/32 is local host.
O IA 192.168.3.0/24 [110/2] via 192.168.2.2, 00:02:49, GigabitEthernet 0/3
```

In this case, after the virtual link is configured, Router B has successfully learned the routes of the network segment 192.168.3.0/24 (Area 2).

- Display the OSPF routing table of Router D.

```
D#show ip route
Codes:  C - connected, S - static, R - RIP, B - BGP
```

```
O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default
Gateway of last resort is no set
C    4.4.4.0/24 is directly connected, Loopback 0
C    4.4.4.4/32 is local host.
O IA 192.168.1.0/24 [110/3] via 192.168.3.1, 00:04:45, GigabitEthernet 0/5
O IA 192.168.2.0/24 [110/2] via 192.168.3.1, 00:05:02, GigabitEthernet 0/5
C    192.168.3.0/24 is directly connected, GigabitEthernet 0/5
C    192.168.3.2/32 is local host.
```

In this case, after the virtual link is configured, Router D has successfully learned the routes of the network segments 192.168.1.0/24 (Area 0) and 192.168.2.0/24 (Area 1).

- Display the OSPF routing table of Router A.

```
A#show ip route ospf
O IA 192.168.2.0/24 [110/2] via 192.168.1.2, 00:51:22, GigabitEthernet 0/1
O IA 192.168.3.0/24 [110/3] via 192.168.1.2, 00:07:58, GigabitEthernet 0/1
```

In this case, after the virtual link is configured, Router A has successfully learned the routes of the network segment 192.168.3.0/24 (Area 2).

- Display the OSPF virtual link information about Router B.

```
B#show ip ospf 1 virtual-links
Virtual Link VLINK0 to router 3.3.3.3 is up
  Transit area 0.0.0.1 via interface GigabitEthernet 0/3
  Local address 192.168.2.1/32
  Remote address 192.168.2.2/32
  Transmit Delay is 1 sec, State Point-To-Point,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:03
Adjacency state Full
```

- Display the OSPF virtual link information about Router C.

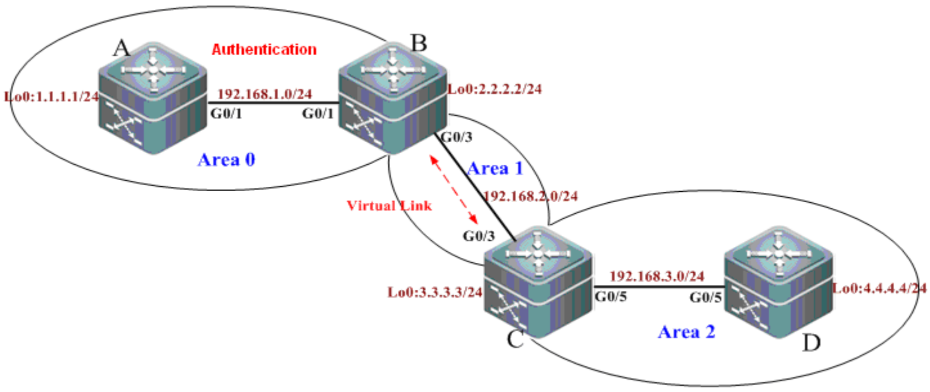
```
C#show ip ospf 1 virtual-links
Virtual Link VLINK0 to router 2.2.2.2 is up
  Transit area 0.0.0.1 via interface GigabitEthernet 0/3
  Local address 192.168.2.2/32
  Remote address 192.168.2.1/32
  Transmit Delay is 1 sec, State Point-To-Point,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:00
    Adjacency state Full
```

Example of OSPF Authentication Configuration

Networking Topology

The following figure shows an OSPF routing area. The network segment 192.168.1.0 belongs to Area 0. The network segment 192.168.2.0 belongs to Area 1. The network segment 192.168.3.0 belongs to Area 2. Due to the limitation of network structures, Area 2 is connected to Area 0 through virtual links.

Networking topology for OSPF authentication configuration



Applications

1. To prevent the device from learning unauthenticated and invalid routes and advertising valid routes to unauthenticated devices, it is required to configure area authentication in the backbone area (Area 0), with the authentication type being MD5.

2. Router D shall be able to learn routes of the network segments 192.168.1.0/24 (Area 0) and 192.168.2.0/24 (Area 1). Meanwhile, Router B shall be able to learn the routes of the network segment 192.168.3.0/24 (Area 2).

Configuration Tips

To configure the OSPF area authentication, configure the area authentication on all routers in the same area with the same authentication type. This example enables the area authentication in Area 0, namely all routers (Router A and Router B) in Area 0 shall be configured with the same authentication type.

When OSPF virtual links are used to connect a non-backbone area (Area 2) with a backbone area, if ID authentication is enabled in the backbone area (Area 0), the identity authentication shall also be configured on the ABR (Router C) in the non-backbone area.

Tips for configuring the OSPF area authentication are shown as follows:

8) In OSPF route configuration mode, specify the authentication type for the area.

9) Configure the authentication type and key on the interface.

Configuration Steps

The following information only shows how to configure the OSPF area authentication. For other configurations, see "Example of OSPF Virtual Link Configuration".

- **Configure Router A.**

Step 1: In OSPF route configuration mode, specify Area 0 to enable the MD5 authentication.

```
A(config)#router ospf 1
A(config-router)#area 0 authentication message-digest
A(config-router)#exit
```

Step 2: Configure the authentication type and key on the interface.

```
A(config)#interface gigabitEthernet 0/1
A(config-if-GigabitEthernet 0/1)#ip ospf message-digest-key 1 md5 hello
```

- **Configure Router B.**

Step 1: In OSPF route configuration mode, specify Area 0 to enable the MD5 authentication.

```
B(config)#router ospf 1
B(config-router)#area 0 authentication message-digest
B(config-router)#exit
```

Step 2: Configure the authentication type and key on the interface.

```
B(config)#interface gigabitEthernet 0/3
B(config-if-GigabitEthernet 0/3)#ip ospf message-digest-key 1 md5 hello
```

- **Configure Router C.**

! Enable the identity authentication of the backbone area (Area 0) on Router C.

```
C(config)#router ospf 1
C(config-router)#area 0 authentication message-digest
```

Verification

Step 1: Display the OSPF information about the routers when the authentication is enabled only on Router A and Router B. (disabled on Router C)

! Display the virtual link configurations of Router B.

```
B#show ip ospf virtual-links
Virtual Link VLINK0 to router 3.3.3.3 is up
  Transit area 0.0.0.1 via interface GigabitEthernet 0/3
  Local address 192.168.2.1/32
  Remote address 192.168.2.2/32
  Transmit Delay is 1 sec, State Point-To-Point,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:09
Adjacency state Down
```

In this case, the adjacency state is down.

! Display the virtual link configurations of Router C.

```
C#show ip ospf virtual-links
Virtual Link VLINK0 to router 2.2.2.2 is up
  Transit area 0.0.0.1 via interface GigabitEthernet 0/3
  Local address 192.168.2.2/32
  Remote address 192.168.2.1/32
  Transmit Delay is 1 sec, State Point-To-Point,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:08
Adjacency state Down
```

In this case, the adjacency state is down.

! Display the OSPF routing information about Router A.

```
A#show ip route ospf
O IA 192.168.2.0/24 [110/2] via 192.168.1.2, 00:10:59, GigabitEthernet 0/1
```

In this case, Router A has failed to learn the routes of Area 2.

Step 1: Display the OSPF information about the routers after the authentication is enabled on Router A and Router B and the identity authentication of Area 0 is enabled on Router C.

! Display the virtual link configurations of Router B.

```
B#show ip ospf virtual-links
Virtual Link VLINK0 to router 3.3.3.3 is up
  Transit area 0.0.0.1 via interface GigabitEthernet 0/3
  Local address 192.168.2.1/32
  Remote address 192.168.2.2/32
  Transmit Delay is 1 sec, State Point-To-Point,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:01
Adjacency state Full
```

In this case, the adjacency state is full.

! Display the virtual link configurations of Router C.

```
C#show ip ospf virtual-links
Virtual Link VLINK0 to router 2.2.2.2 is up
Transit area 0.0.0.1 via interface GigabitEthernet 0/3
Local address 192.168.2.2/32
Remote address 192.168.2.1/32
Transmit Delay is 1 sec, State Point-To-Point,
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
Hello due in 00:00:00
Adjacency state Full
```

In this case, the adjacency state is full.

! Display the OSPF routing information about Router A.

```
A#show ip route ospf
O IA 192.168.2.0/24 [110/2] via 192.168.1.2, 00:21:30, GigabitEthernet 0/1
O IA 192.168.3.0/24 [110/3] via 192.168.1.2, 00:03:18, GigabitEthernet 0/1
```

In this case, Router A has successfully learned the routes of Area 2.

Step 3: Display general OSPF information about Router A.

```
A#show ip ospf
 Routing Process "ospf 1" with ID 1.1.1.1
 Process uptime is 18 hours 22 minutes
 Process bound to VRF default
 Conforms to RFC2328, and RFC1583 Compatibility flag is enabled
 Supports only single TOS(TOS0) routes
 Supports opaque LSA
 Enable two-way-maintain
 Initial SPF schedule delay 1000 msecs
 Minimum hold time between two consecutive SPFs 5000 msecs
 Maximum wait time between two consecutive SPFs 10000 msecs
 Initial LSA throttle delay 0 msecs
 Minimum hold time for LSA throttle 5000 msecs
 Maximum wait time for LSA throttle 5000 msecs
 Lsa Transmit Pacing timer 40 msecs, 10 LS-Upd
 Minimum LSA arrival 1000 msecs
 Pacing lsa-group: 240 secs
 Number of incoming current DD exchange neighbors 0/5
 Number of outgoing current DD exchange neighbors 0/5
 Number of external LSA 0. Checksum 0x000000
 Number of opaque AS LSA 0. Checksum 0x000000
 Number of non-default external LSA 0
 External LSA database is unlimited.
 Number of LSA originated 2
 Number of LSA received 244
 Log Neighbor Adjacency Changes: Enabled
  Number of areas attached to this router: 1: 1 normal 0 stub 0 nssa
    Area 0 (BACKBONE)
        Number of interfaces in this area is 1(1)
        Number of fully adjacent neighbors in this area is 1
        Area has message digest authentication
        SPF algorithm last executed 17:24:38.030 ago
        SPF algorithm executed 11 times
        Number of LSA 7. Checksum 0x032955
```

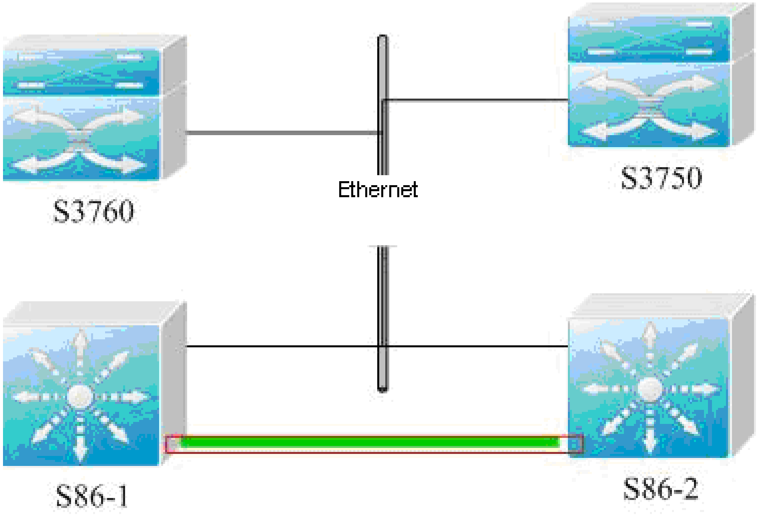
The above information shows that the area authentication has been enabled.

Example of OSPF GR Configuration

Configuration Requirements

The following figure shows that two S86 high-end switches have the GR Restart capability and are equipped with primary and secondary engines to support redundant backup at the control plane. S86-1 establishes the OSPF adjacency relationship with S86-2, S3760, and S3750. The OSPF GR capability is supported by all devices. The connection layout is shown as follows:

OSPF GR configuration



It is required that two S86 devices shall support non-stop packet forwarding to enhance the reliability of core devices.

Specific Configurations

Configure S3760.

```
Ruijie(config)# router ospf 1
Ruijie(config-router)# graceful-restart helper strict-lsa-checking
```

Configure S3750.

```
Ruijie(config)# router ospf 1
Ruijie(config-router)# graceful-restart helper strict-lsa-checking
```

Configure S86-1.

```
Ruijie(config)# router ospf 1
Ruijie(config-router)# graceful-restart
Ruijie(config-router)# graceful-restart helper strict-lsa-checking
```

Configure S86-2.

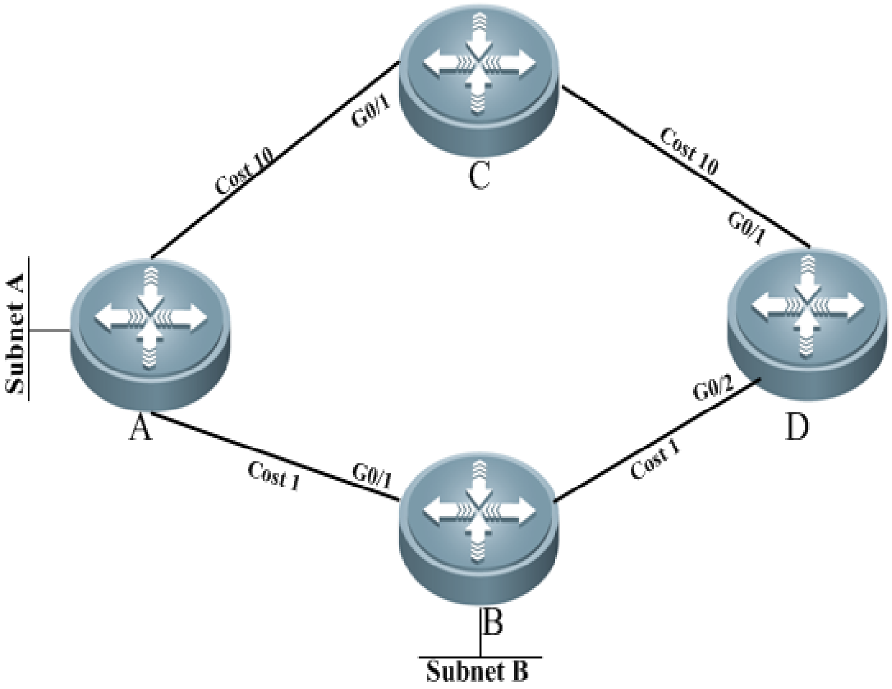
```
Ruijie(config)# router ospf 1
Ruijie(config-router)# graceful-restart
Ruijie(config-router)# graceful-restart helper strict-lsa-checking
```

Example of OSPF Stub Router Configuration

Configuration Requirements

Four routers form an OSPF routing area. The connection layout is shown in the following figure. According to the rule for optimal routing, the route from D to subnet A passes B. It is expected that the route passes C by changing configurations of B only.

OSPF Stub Router configuration



It is required that B only transmit routes to Subnets B and C transmit other routes.

Specific Configurations

Configure IP addresses and OSPF processes on the four routers, and make the following configurations after the adjacency relationship have been established successfully.

Configurations on D:

# Configure the Ethernet interface.

```
interface gigabitEthernet 0/1
ip ospf cost 10
interface gigabitEthernet 0/2
ip ospf cost 1
```

Configurations on C:

# Configure the Ethernet interface.

```
interface gigabitEthernet 0/1
ip ospf cost 10
```

Configurations on B:

# Configure the Ethernet interface.

```
interface gigabitEthernet 0/1
ip ospf cost 1
```

Configure the OSPF routing protocol.

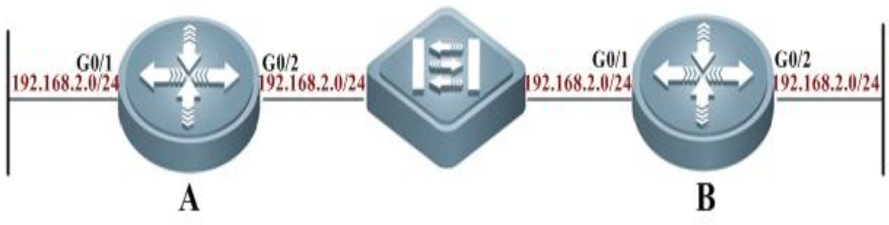
```
router ospf 1
max-metric router-lsa
```

Example of OSPF Fast Convergence Configuration

Configuration Requirements

Routers A and B are interconnected through a layer-2 switch. Run the OSPF protocol on them to establish routes. The following figure shows details about IP address assignment and connection layout.

OSPF fast convergence configuration



After link failure between B and the layer-2 switch occurs, A shall be able to detect adjacency changes within 1 second and quickly respond to the network changes.

Specific Configurations

The Fast Hello function reduces the time for detecting adjacency changes to less than 1 second. Meanwhile, the LSA fast convergence function facilitates to adaptation to the swift network changes.

Configurations on A:

# Configure the Ethernet interface.

```
interface gigabitEthernet 0/1
ip address 192.168.1.1 255.255.255.0
interface gigabitEthernet 0/2
ip address 192.168.2.1 255.255.255.0
ip ospf dead-interval minimal hello-multiplier 5
```

# Configure the OSPF routing protocol.

```
router ospf 1
network 192.168.1.0 0.0.0.255 area 0
network 192.168.2.0 0.0.0.255 area 0
```

```
timers arrival-time 100
timers throttle lsa all 0 100 500
```

Configurations on B:

# Configure the Ethernet interface.

```
interface gigabitEthernet 0/1
ip address 192.168.3.1 255.255.255.0
interface gigabitEthernet 0/2
ip address 192.168.2.2 255.255.255.0
ip ospf dead-interval minimal hello-multiplier 5
```

# Configure the OSPF routing protocol.

```
router ospf 1
network 192.168.2.0 0.0.0.255 area 0
network 192.168.3.0 0.0.0.255 area 0
timers arrival-time 100
timers throttle lsa all 0 100 500
```

Configuring OSPFv3

OSPFV2 (RFC2328, OSPFv2) runs under the IPv4. The RFC5340 describes OSPFV3, the extension of OSPFv2 that provides support for IPv6 routes. This document briefly describes the OSPFv3 protocol and its configuration.

Before learning this document, you must know the OSPFv2 protocol and related configuration.

The OSPFv3 protocol extends the OSPFv2 protocol with the main operating mechanisms and most configuration the same as the OSPFv2.

Overview

As an Interior Gateway Protocol (IGP), the OSPF runs among the layer 3 devices within an Autonomous System (AS).

Unlike a vector distance protocol, the OSPF is a link-state protocol. By exchanging various types of link-state advertisements (LSAs) recording link state between devices, it synchronizes link state information between devices and then calculates OSPF route entries through the Dijkstra algorithm.

The OSPFv3 is described in the RFC5340 and supports the IPv6. This section describes the differences from the OSPFv2 in implementation.

LSA Association Change
Interface Configuration
Router ID Configuration
Authentication Mechanism Configuration

LSA Association Change

Just as described above, the OSPF is a link-state protocol and its implementation is based on LSAs. Through LSAs, we can know the topologies of networks and address information. In contrast to the IPv4, the IPv6 uses 128-bit IP addresses. The design of LSAs is modified accordingly. Firstly, the LSA types are described as follows:

Router-LSAs (Type 1)

Each device generates this type of LSAs by itself. They describe the states of its links in specified areas and the cost spent in reaching the links. In contrast to the OSPFv2, the Router-LSAs of the OSPFv3 only indicate the state information of links. They do not record the information about the network addresses connected to routers. The information will be acquired by newly added types of LSAs. Additionally, in the OSPFv2, only one Router-LSA is allowed to be generated for each device in each area. While in the OSPFv3, multiple Router-LSAs are allowed to be generated. Thus, when performing the SPF calculation, we must consider all the Router-LSAs generated by the device. Router-LSAs and Network-LSAs describe the link topology of areas together.

Through the flag bits on Router-LSAs, we can know whether the routers are Area Border Routers (ABR), AS boundary routers (ASBR) or those at one end of a virtual link.

Network-LSAs (Type 2)

Network-LSAs only exist in broadcast networks or NBMA networks and are generated by DRs (Designated Routers) in a network. They describe the information about all the routers connected in specified areas on a network. Like Router-LSAs, Network-LSAs also only indicate the link-state information and do not record the network address information. Network-LSAs and Router-LSAs describe the link topology of areas together.

Inter-Area-Prefix-LSAs (Type 3)

They are generated for an area by the ABRs in the area and used to describe the network information about reaching other areas. They replace type 3 summary-LSAs in OSPFv2. In contrast to the OSPFv2, they use a prefix structure to describe the destination network information.

Inter-Area-Router-LSAs (Type 4)

They are generated for an area by the ABRs in the area, used to describe the path information about reaching the ASBRs in other areas, and replace type 4 summary-LSAs in OSPFv2.

AS-external-LSAs (Type 5)

This type of LSAs is generated by ASBRs and used to describe the network information about reaching outside the AS. Usually, the network information is generated through other routing protocols. In contrast to the OSPFv2, it uses a prefix structure to describe the destination network information.

NSSA-LSA (Type 7)

Their function is same as that of type 5 AS-external-LSAs. However, they are generated by ASBRs in the NSSA area.

Link-LSAs (Type 8)

In the OSPFv3, the newly added LSA type is generated by each device for each connected link and describes the local link address of the device in the current link and all set IPv6 address prefix information.

Intra-Area-Prefix-LSAs (Type 9)

It is a newly added LSA type in the OSPFv3 and provides additional address information for Router-LSAs or Network-LSAs. Therefore, it plays two roles:

- Associating network-LSAs and recording the prefix information of a transit network.
- Associating router-LSAs and recording the prefix information on all Loopback interfaces, point-to-point links, point-to-multipoint links, virtual links and stub networks of the router in the current area.

Other major changes associated with LSA:

LSA flooding scope

In the OSPFv2, the LSA flooding occurs inside areas and ASs. In the OSPFv3, flooding occurs also in local links. Type 8 Link-LSAs is the type that can flood only inside a local link.

Handling an unknown LSA type

This is an improvement of OSPFv3 based on OSPFv2.

In the OSPFv2, database synchronization is necessary in the initial establishment of the adjacency relationship. If there is an unrecognizable LSA type in the database description message, this relationship cannot be established properly. If there is an unrecognizable LSA type in a link-state updating message, then the type of LSAs will be dropped.

In the OSPFv3, it is allowed to receive an unknown LSA type. By using the information recorded in the LSA header, we can determine how to handle the unrecognizable LSA type received.

Interface Configuration

In the OSPFv3, the changes based on interface configuration are as follows:

In order for an interface to run OSPFv3, enable the OSPFv3 directly in the interface configuration mode. For OSPFv2, however, run the **network** command in the OSPF route configuration mode.

If an interface runs OSPFv3, all the addresses on the interface will run IPv6. In the OSPFv2,however, all the addresses are enabled via the **network** command.

In the environment where the OSPFv3 runs, a link can support multiple OSPF entities and different devices connecting this link can run one of these OSPF instances. The OSPFv3 adjacency can only be established between the devices with the same instance ID. The OSPFv2 does not support this function.

Router ID Configuration

RFC5340 specifies the OSPFv3 Router ID is in the format of 32-bit IPv4 address but not the IPv6 address.

By default, the methods of electing OSPFv3 Router ID are the same as the OSPFv2 process . The automatic election method is adopted. Firstly, the largest IPv4 address for the loopback interface is elected as the Router ID. if the loopback interface of IPv4 addresses has not been configured, OSPFv3 process will select the largest IPv4 address for other interfaces as the Router ID. With multiple OSPFv3 processes running on the device, the OSPFv3 process selects the Router ID with the highest priority from the unselected IPv4 addresses in the above way. Different Router IDs are for the different processes.

If the IPv4 addresses available for the Router ID selection are insufficient, the OSPFv3 process will fail to auto-obtain the Router ID. You can use the **router-id** command to configure a Router ID to enable the OSPFv3 process.

The Router ID for each router in the AS must be unique. With multiple OSPFv3 processes running on the same device, the Router ID for each process must also be unique.

Authentication Mechanism Configuration

OSPFv2 itself supports two authentication modes: plain text authentication and key authentication based on MD5. Authentication fields have been removed from OSPFv3 packet headers. OSPFv3 does away its support for authentication entirely, instead relying on IPsec framework offered by IPv6. When OSPFv3 runs on IPv6, OSPFv3 requires the IPv6 authentication header (AH) or IPv6 encapsulating security payload (ESP) to ensure integrity and confidentiality of routing exchanges.

Configure authentication commands to enable IP AH, which only provides authentication on data integrity and consistency. Configure encryption commands to enable IP ESP, which covers AH functions and ensures confidentiality. Namely, authentication and encryption are performed simultaneously.

OSPFv3 authentication configuration can be based on an interface, an area or a virtual link. If you want to achieve higher security, configure different IPsec authentications on every interface. Area configuration is effective for all interfaces except the virtual link within this area. If area configuration and interface configuration are both performed, IPsec of interface configuration has a higher priority.

Basic Configuration of OSPFv3

The OSPFv3 protocol of Ruijie Network has the following features:

Supports multi-instance OSPF;

Supports network type setting;

Supports virtual links;

Supports passive interfaces;

Supports an interface to select a participant OSPF instance;

Supports stub area;

Supports route redistribution;

Supports route aggregation;

Supports timer setting;

Supports link detection using BFD mechanism

To be implemented:

Supports NSSA areas;

Supports authentication. The OSPFv3 will use the IPSec authentication mechanism.

Default OSPFv3 Configuration:

Router ID		Undefined
Interface Configuration	Interface type	Broadcast network
	Interface cost	Undefined
	Hello message sending interval	10 seconds
	Dead interval of adjacent device	4 times of the hello interval.
	LSA sending delay	1 seconds
	LSA retransmitting interval	5 seconds
	Priority	1
	MTU check of database description messages	Enabled
Virtual Link	Virtual Link	Undefined
	Hello message sending interval	10 seconds
	Dead interval of adjacent device	4 times the hello interval.
	LSA sending delay	1 seconds
	LSA retransmitting interval.	5 seconds
	Fast Hello function	Disabled
Area Configuration	Area	Undefined
	Default router cost for stub areas	1
Routing information Aggregation	Inter-area route aggregation	Off
	External route aggregation	Off
Management Distance	Intra-area route	110
	Inter-area route	110
	External route	110
Auto cost generation		Enabled The default cost reference is 100 Mbps.

Shortest path first (SPF) timer	Time from receiving the topology change to running next SPF calculation :5 seconds The least interval between two calculations: 10 seconds
Route redistribution	Off
Route filtering	Off
Passive interface	Off

Enabling OSPFv3

Perform the following steps in the privileged mode to enable the OSPFv3:

Command	Function
configure terminal	Enter the global configuration mode
ipv6 router ospf *process-id*	Start an OSPFv3 routing process and enter the OSPFv3 configuration mode.
router-id *router-id*	Configure the Router ID used for OSPFv3 running on this device.
interface interface-id	Enter the interface configuration mode
ipv6 ospf process-id **area** *area-id* **[instance** *instance-id***]**	Enable the OSPFv3 on the interface. instance-id：Set the interface instance ID that participates the OSPFv3. The interfaces of different devices connected to the same network can choose different OSPFv3 instances to participate.
copy running-config startup-config	Save the configuration.

The OSPFv3 instance ID and process ID are different. OSPFv3 process ID is valid for the device itself only, not influencing the interaction with other routers. While the OSPFv3 instance ID influences the interaction with other routers. Only the devices with the same instance ID can set up the OSPFv3 neighbor relationship.

First enable the interface to participate in the OSPFv3 and then configure the OSPFv3 process in the interface configuration mode. Once the process is configured, the interface will automatically participate in the corresponding process. Currently, our products can support up to 32 OSPFv3 processes.

Configuring OSPFv3 Parameters on the Interface

You can modify the interface parameters in the interface configuration mode according to the actual application.

To configure the OSPFv3 interface parameters, execute the following commands in the interface configuration mode:

Command	Function
ipv6 ospf *process-id* **area** *area-id* [**instance** *instance-id*]	Set the interface to participate in the OSPFv3 routing process.
ipv6 ospf network {**broadcast** | **non-broadcast** | **point-to-point** | **point-to-multipoint** [**non-broadcast**]} [**instance** *instance-id*]	Set the network type of an interface. The default is the broadcast network type.
ipv6 ospf neighbor *ipv6-address* {[**cost** *<1-65535>*] | [**poll-interval** *<0-2147483647>* | **priority** *<0-255>*]} [**instance** *instance-id*]	(Optional) Set the OSPFv3 neighbor.
ipv6 ospf cost *cost* [**instance** *instance-id*]	(Optional) Define the interface cost.
ipv6 ospf hello-interval *seconds* [**instance** *instance-id*]	(Optional) Set the interval for sending the Hello packets on the interface. For all nodes in adjacency on the network, this value must be identical.
ipv6 ospf dead-interval *seconds* [**instance** *instance-id*]	(Optional) Set the adjacency dead-interval on the interface. For all nodes in adjacency on the network, this value must be identical.
ipv6 ospf transmit-delay *seconds* [**instance** *instance-id*]	(Optional) Set the delay in sending LSA on the interface.
ipv6 ospf retransmit-interval *seconds* [**instance** *instance-id*]	(Optional) Set the interval for retransmitting LSA on the interface.
ipv6 ospf priority *number* [**instance** *instance-id*]	(Optional) Set the priority of the interface for elect the DR and BDR.
ipv6 ospf authentication ipsec spi *spi* [**md5** | **sha1**] [**0** | **7**] *key*	(Optional) Sets the same interface authentication parameters on both sides.. *spi*: security parameter index within the ragne from 256 to 4294967295. **md5**: specifies md5 authentication mode. **sha1**: specifies sha1 authentication mode. **0**: specifies the key to be displayed as plain text. **7**: specifies the key to bedisplayed as cipher text. *key*: authentication key.

Command	Function
ipv6 ospf encryption ipsec spi *spi* **esp** *null* [**md5** | **sha1**] [**0** | **7**] *key*	(Optional) Sets the same interface authentication parameters on both sides.. *spi*: security parameter index within the ragne from 256 to 4294967295. **null**: specifies null encryption mode. **md5**: specifies md5 authentication mode. **sha1**: specifies sha1 authentication mode. **0**: specifies the key to be displayed as plain text. **7**: specifies the key to bedisplayed as cipher text. *key*: authentication key.

Use the **no** form of the above command to invalidate the configuration.

You can modify the parameter settings according to the actual needs. But note that some parameter settings must be consistent with those of neighbors otherwise no adjacency can be established. These parameters include the following: **instance**, **hello-interval**, **dead-interval**, **authentication** and **encryption**

Configuring OSPFv3 Area Parameter

The OSPF protocol applies the concept of “hierarchical structure”, allowing a network to be divided into a group of parts connected through a “backbone” in a mutual independence way. These parts are called Areas. The backbone part is called Backbone Area and always indicated by the numerical value 0 (or 0.0.0.0).

By using this hierarchical structure, each device is allowed to keep the link state database in the area where it resides and the topology inside the area is invisible to the outside. In this way, the link state database of each device can be always in a reasonable size, the route calculation time is not too much and the number of packets is not too big.

In the OSPF, the following types of special areas have been defined to meet actual needs:

stub Area.

If an area is at the end of the whole network, then we can design the area as a stub area.

A stub area cannot learn the external routing information of an AS (type 5 LSAs). In practical application, external routing information takes a great proportion in the link state database. Therefore, the devices inside a stub area will learn very little routing information, thus reducing the system resources for running the OSPF protocol.

A device inside a stub area can reach outside of an AS through the default route entry (type3 LSA) generated from the default routing information published by Area Border Routers in the stub area.

NSSA area (Not-So-Stubby Area)

NSSA is the extension of the stub area. By preventing from flooding type 5 LSAs to the devices in the NSSA, it reduce the consumption of device resources. However, unlike a stub area, it allows a certain amount of external routing information of the AS to enter an NSSA in other ways, namely, inject into the NSSA in the form of type 7 LSAs.

So far, the NSSA area functions of the OSPFv3 have not be implemented.

To configure the OSPFv3 area parameters, execute the following commands in the OSPFv3 configuration mode:

Command	Function
area *area-id* **stub** [**no-summary**]	Configure a stub area. no-summary: configure the area to a totally stub area, preventing the area border router in the stub area from sending type3 and type4 LSAs to the stub area.
area *area-id* **default-cost** *cost*	Configure the cost of the default route sent to a stub area.

Use the **no** form of the above command to invalidate the configuration.

You can configure the parameter **default-cost** after the configuration of the stub area is made.If the stub area is changed into an ordinary area, the default-cost configuration will be deleted automatically.

Configuring OSPFv3 Virtual Link

In the OSPF, all areas must be connected to the backbone area to ensure the communication with other areas. If some areas cannot be connected to the backbone area, virtual links are required to connect the backbone area.

To establish a virtual link, execute the following commands in the OSPFv3 configuration mode:

Command	Function
area *area-id* **virtual-link** *router-id* [**hello-interval** *seconds*] [**dead-interval** *seconds*] [**transmit-delay** *seconds*] [**retransmit-interval** *seconds*] [**instance** *instance-id*] [**authentication ipsec spi** *spi* [**md5** | **sha1**] [**0** | **7**] *key*] [**encryption ipsec spi** *spi* **esp null** [**md5** | **sha1**] [**0** | **7**] *key*]	Configure a virtual link. By default, the virtual link is not configured. *area-id:* the ID for the area where the virtual link is. *router-id:* the router-id for the virtual link neighbor. *spi*: security parameter index within the ragne from 256 to 4294967295. **null**: specifies null encryption mode. **md5**: specifies md5 authentication mode. **sha1**: specifies sha1 authentication mode. **0**: specifies the key to be displayed as plain text. **7**: specifies the key to bedisplayed as cipher text. *key*: authentication key. Other parameters have the same meanings as the interface parameters.

Use the **no** form of the command to invalidate the configuration.

It is not allowed to create a virtual link in the stub area .A virtual link can be taken as a special interface, so its configuration are the same as that of a normal interface. You must ensure that the configuration of **instance, hello-interval**, **dead-interval**, **authentication** and **encryption** configured at the two ends of the virtual link are identical.

Configuring OSPFv3 Route Aggregation

Without route aggregation, each device on the network has to maintain the routing information to every network.By aggrgating some information together, route aggregation can alleviate the burden on the L3 device and network bandwidth. As the size of a network is growing, the importance of route aggregation increases..

Ruijie's L3 devices support two types of route aggregation configuration: inter-area route aggregation and external route aggregation.

Configuring Inter-area Route Aggregation

The ABR in an area needs to advertise the routes in an area to other areas. If the route addresses are continual, the ABR aggregates the routing information and then advertises it.

To configure the inter-area route aggregation, execute the following commands in the OSPFv3 configuration mode:

Command	Function
area *area-id* **range** *ipv6-prefix*/*prefix-length* [**advertise** | **not-advertise**]	Configure inter-area route aggregation. *area-id*: ID of the area for aggregation. *ipv6-prefix/prefixlength*: Set the ipv6 prefix of the aggregated route. **advertise|not-advertise**: Advertise the summary-LSA created by aggregation or not.

Use **no area** *area-id* **range** *ipv6-prefix /prefix-length* to remove the inter-area aggregation configured.

Configuring External Route Aggregation

The route aggregation is allowed when redistributing the generated Type-5 LSA on the ASBR.

To configure the external route aggregation, execute the following commands in the OSPFv3 configuration mode:

Command	Function
summary-prefix *ipv6-prefix* / *prefix-length* [**not-advertise** | **tag** *tag-value*]	Configure external route aggregation. *ipv6-prefix/prefixlength*: Set the ipv6 prefix of the aggregated route. **not-advertise**: Not advertise the LSA created by aggregation. *tag-value*: The valid range is *<0-4294967295>*, used to specify the tag value for the LSA created by aggregation.

Use **no summary-prefix** *ipv6-prefix*/*prefix-length* to remove the external route aggregation configured.

Configuring Bandwidth Reference Value of OSPFv3 Interface Metric

The metric for the OSPF protocol is a bandwidth value based on the interface. The cost value of the interface is calculated based on its bandwidth.

For example, if the bandwidth reference value of an interfaces is 100 Mbps and the bandwidth of the network interfaces is 10Mbps, the automatically calculated interface cost is 100/10=10.

Currently, the default reference value of the network interface bandwidth of our products is 100Mbps.

To modify the reference value of the OSPFv3 interface bandwidth, execute the following commands in the OSPFv3 configuration mode:

Command	Function
auto-cost [**reference-bandwidth** *ref-bw*]	Configure the bandwidth reference value for interface metric, in Mbps.

You can run the **ipv6 ospf cost** *cost-value* command in the interface configuration mode to set the cost for a specified interface, which takes precedence over the one calculated based on bandwidth reference value.

Configuring MTU Check of DD Packets Received on OSPFv3 Interfaces

When the OSPFv3 receives the DD(Database Description) packets, it checks whether the MTU for the neighbor interface is the same as the MTU for its own interface. If the former is larger than the latter, the adjacency relationship cannot be established.

By default, this function of MTU check is disabled. To enable the MTU check on an interface, execute the following command in the interface configuration mode:

Command	Function
no ipv6 ospf mtu-ignore [**instance** *instance-id*]	Enable the MTU check on the interface when receiving database description (DD) packets.

By default, the MTU check function of the interface is disabled.

Configuring OSPFv3 Default Route

In the OSPFv3 protocol, the default route can be generated in many ways.

As described in section "Configuring OSPFv3 Area Parameters", the default route represented by Type-3 LSA will be automatically generated in a stub area.

You can configure a default route represented by Type 5 LSA and advertise it to the whole OSPF AS.

Execute the following command in the OSPFv3 configuration mode:

Command	Function
default-information originate [always] [metric *metric-value***] [metric-type** *type-value***] [route-map** *map-name***]**	Configure the generation of a default route. **always:** With this parameter configured, no matter what the condition the system routing is in, a default route LSA is always generated. With this parameter not configured, only when the default routing existd in the core routing table, the default route LSA is generated and advertised. *metric:* Initial metric value of the route. The valid range is 0-16777214. **metric-type:** The external routing type corresponding to the default routing. **route-map:** the corresponding route-map rule to set the generated LSA.

Use **no default-information originate** to remove the default routing generated.

This command cannot be configured on the devices in a stub area.

Once configured, the device automatically becomes the ASBR.

Configuring OSPFv3 Routing Redistribution

Routing information redistribution allows the routing information of a routing protocol to be redistributed to another routing protocol.

To configure the OSPFv3 route redistribution, execute the following commands in the OSPFv3 configuration mode:

Command	Function
redistribute {**bgp** | **connected** | **isis** [*area-tag*] | **ospf** *process-id* | **rip** | **static**} [{**level-1** | **level-1-2** | **level-2**} | **match** {**internal** | **external** [**1**|**2**]} | **metric** *metric-value* | **metric-type** {*1*|*2*} | **route-map** *route-map-name* | **tag** *tag-value*]	Redistribute the routing information of other routing protocols. And set the conditions of redistribution. At present, the OSPFv3 supports redistribution of static, connect, rip, bgp, isis and ospf routes. When redistributing ISIS routes, you can configure the level parameter to redistribute the ISIS routes at the specified level. When redistributing OSPF routes, you can configure the match parameter to redistribute the OSPF routes of the specific sub type.
default-metric *number*	Configure the default metric for route redistribution.

Use the **no** redistribute *protocol* to disable the routing information redistribution.

- The isis parameter is not supported by S8600 and S12000 series in v10.4(3b17).

Configuring OSPFv3 Timer

After receiving the notice of network topology changes, the OPSPFv3 routing process will wait for a period of time before starting the SPF calculation. The SPF calculation delay is configurable, you can also use the command to configure the minimum and maximum interval between two SPF calculations.

To configure the OSPFv3 routing calculation timer, execute the following command in the routing process configuration mode:

Command	Function
timers throttle spf *spf-delay spf-holdtime spf-max-waittime*	Configure the OSPFv3 timer of routing calculation, in ms.

The parameter *spf-delay* refers to the delay from the topology change to the beginning of the SPF calculation.

The parameter *spf-holdtime* refers to the minimum interval of the first and the second SPF calculations triggered. Afterwards, the next SFP holdtime shall at least be twice as the last interval till the interval reaches the configured *spf-max-waittime*. If the SPF calculation intervals have exceeded the minimum value, it will re-calculate the SPF calculation interval from the *spf-holdtime.*

In normal conditions, when the link changes occassionally, reducing the *spf-delay* and *spf-holdtime* value can speed up the OSPF convergence. Setting a large *spf-max-waittime* avoids high CPU consumption by OSPF due to the continuous link fluctuation.

For example, **timers throttle spf** *1000 5,000 100,000*

If the topology keeps changing, the SPF calculation intervals(the SPF calculation interval increases by the binary exponential backoff algorithm, but cannot exceed the max-wait-time) are 1s, 6s, 16s, 36s, 76s, 156s, 256s, 256+100, ……

To configure the delay and holdtime for OSPFv3 routing calculation only, execute the following command in the routing process configuration mode:

Command	Function
Ruijie (config-router)# timers spf *spf-delay spf-holdtime*	Configure the routing calculation timer in second.

The **timers spf** and **timers throttle spf** commands are overwritten, and the one configured later is valid. With both commands not configured, the default value is **timers throttle spf.**

The function of the command **timers throttle spf** has covered the function of **timers spf**, and is even stronger. It is recommended to use the **timer throttle spf** command.

Configuring OSPFv3 Passive Interface

To prevent other Layer 3 devices in the network from learning the routing information of this device, you can set a network interface to a passive interface in the routing protocol configuration mode

For the OSPFv3 protocol, if a network interface is configured as a passive network interface, then this network interface will receive/send no OSPF message.

To configure an OSPFv3 passive interface, execute the following command in the OSPFv3 configuration mode:

Command	**Function**
passive-interface {**default** | *interface-type interface-number* }	Configure a passive interface. default: with this parameter configured, all interfaces will be set as the passive interfaces. Interface: set the specified interface as the passive interface. Combining **passive-interface default** and **no passive-interface *interface*** can specify some interfaces as non-passive interfaces and the others as passive interfaces.

Use the command **no passive-interface** {*interface-id* | **default**} to remove the passive interface setting.

Configuring OSPFv3 Authenticated Encryption

Authenticated encryption is configured to avoid learning unauthenticated encryption and invalid routes, and prevent valid routes from being announced to the unauthenticated encryption device. On a broadcast network, authenticated encryption can also help avoid the possibility of specifying the unauthenticated encryption device to ensure stability and invasion-resistance of the routing system.

Run the following demands to configure OSPFv3 authenticated encryption in routing process configuration mode or interface configuration mode:

Command	**Function**
area *area-id* authentication ipsec spi *spi* [**md5** | **sha1**] [**0** | **7**] *key*	Enables authentication on the area. Sets authentication mode and key.
area *area-id* **encryption ipsec spi** *spi* **esp null** [**md5** | **sha1**] [**0** | **7**] *key*	Enables authenticated encryption on the area. Sets encrypted authentication mode and key.
ipv6 ospf authentication ipsec spi *spi* [**md5** | **sha1**] [**0** | **7**] *key*	Enables authentication on the interface. Sets authentication mode and key.
ipv6 ospf encryption ipsec spi *spi* **esp null** [**md5** | **sha1**] [**0** | **7**] *key*	Enables authenticated encryption on the interface. Sets encrypted authentication mode and key.

Use the **no** form of this command to disable configuration.

Note

Connected interfaces within the same area must be configured with the same authenticated encryption parameters. Authenticated encryption configured on the area is effective for all interfaces (except the virtual link) within this area but authenticated encryption configured on the interface has a higher priority.

Authenticated encryption parameters configured on two ends of the virtual link must be the same. All *spi* parameters must be unique.

Please refer to *configuring OSPFv3 virtual link* section for virtual link authentication configuration.

Configuring the OSPFv3 Route Management Distance

The route management distance, representing the reliability of the route source, is used to compare the priorities for different routing protocols. The valid range for the management distance is 0-255. The smaller the management distance is, the higher the route priority is, and the higher the route source reliability is.

By default, the OSPFv3 route management distance is 110. You can set different management distances for different OSPFv3 routes, the intra-area, inter-area and external routes.

To change the OSPFv3 route management distance, execute the following command in the routing process configuration mode:

Command	**Function**
distance {*distance* | **ospf** { **intra-area** *distance* | **inter-area** *distance* | **externa**l *distance* }}	Modify the OSPFv3 route management distance.

The management distance must be used to compare the priorities of the different routes originated from different OSPFv3 processes.

Configuring the OSPFv3 BFD

Refer to relevant sections in BDF Configuration Guide for the OSPFv3 BFD configuration.

OSPFv3 Debugging & Monitoring

OSPFv3 supports a large range of debugging and monitoring commands.

OSPFv3 Debugging Commands

Use the following commands to enable the OSPFv3 process debugging in the privileged configuration mode:

Command	Function
debug ipv6 ospf events	Show the OSPFv3 event information.
debug ipv6 ospf ifsm	Show the state machine events and changes of an egress interface.
debug ipv6 ospf lsa	Show the related OSPFv3 LSA information.
debug ipv6 ospf nfsm	Show state machine events and changes of a neighbor.
debug ipv6 ospf nsm	Show the related OSPFv3 and NSM module information.
debug ipv6 ospf packet	Show the OSPFv3 packet information.
debug ipv6 ospf route	Show the OSPF routing calculation and addition information.

Use the **undebug** form of the above commands to disable the above **debug** commands.

The **debug** commands are provided for technicians.

Running a **debug** command will affect the performance of the system in a certain extent. Therefore, after running a **debug** command, use an **undebug** command to disable the debug command.

OSPFv3 Monitoring Commands

Use the following commands to enable the OSPFv3 process monitoring in the privileged configuration mode:

Command	Function
show ipv6 ospf	Show the OSPFv3 process information.
show ipv6 ospf [*process-id*] **database** [**isa-type** [**adv-router** *router-id*]]	Show the database information of the OSPF process.
show ipv6 ospf interface *[interface-type interface-number]*	Show the interface information of the OSPFv3 process.
show ipv6 ospf [*process-id*] **neighbor** [*interface-type interface-number* [**detail**]] *[neighbor-id*] [**detail**]	Show the neighbor information of the OSPFv3 process.
show ipv6 ospf [*process-id*] **route**	Show the OSPFv3 routing information.
show ipv6 ospf [*process-id*]**summary-prefix**	Show the OSPFv3 external route summary information
show ipv6 ospf [*process-id*] **topology** [**area** *area-id*]	Show each area topology of the OSPFv3.
show ipv6 ospf [*process-id*] **virtual-links**	Show the virtual link information of the OSPFv3 process.

OSPFv3 Configuration Examples

OSPFv3 Basic Configuration Example

The following configuration example shows the commands related to OSPF configuration.

Topological Diagram

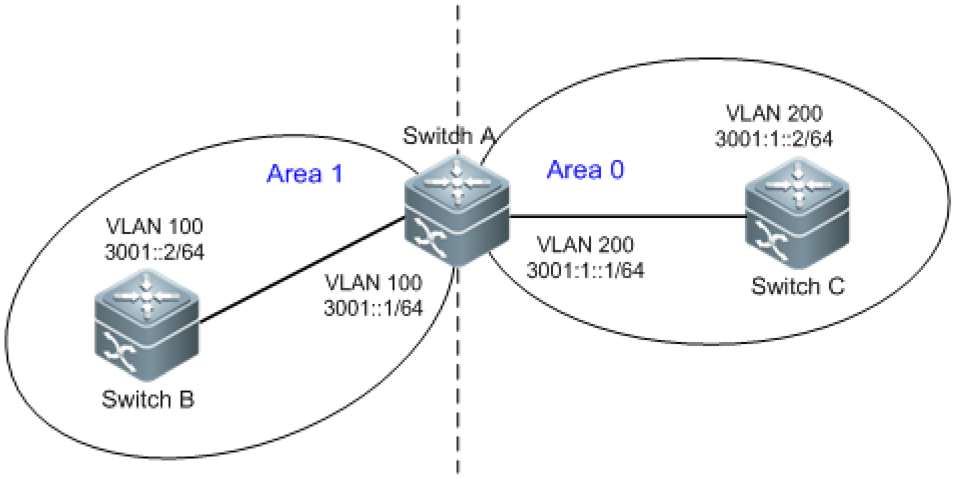


Figure 1 OSPFv3 basic configuration

SwitchA and SwitchB belong to Area 0, while Switch A and Switch C belong to Area 1. The intercommunication between three switches is realized via the vlan interface.

Application Requirements

Enable the OSPFv3 on all switches and divide them into two areas between which IPv6 packets can be communicated.

Configuration Tips

Key points

Configure Area0 and Area1, and enable OSPFv3 on the corresponding VLAN interface of the switch (interface vlan 100 or vlan 200 of SwtichA/SwitchB/SwitchC in this example)

Cautions

The router-id must be specified, or else the adjacency cannot be created. Automatic acquisition of router-id is supported in 10.4 and subsequent releases.

Vlan must be created first, or else the VLAN interface cannot join OSPFv3.

Configuration Steps

Configuring SwitchA

Step 1, Create a VLAN and set the IPv6 address

```
SwitchA# config terminal
```

Enter configuration commands, one per line.   End with CNTL/Z.

! Create and configure interface vlan100

```
SwitchA(config)# vlan 100
SwitchA(config-vlan)#exit
SwitchA(config-vlan)#interface vlan 100
SwitchA(config-if-VLAN 100)#ipv6 enable
SwitchA(config-if-VLAN 100)#ipv6 address 3001::1/64
SwitchA(config-if-VLAN 100)#exit
```

! Create and configure interface vlan200

```
SwitchA(config)#vlan 200
SwitchA(config-vlan)#interface vlan 200
SwitchA(config-if-VLAN 200)#ipv6 enable
SwitchA(config-if-VLAN 200)#ipv6 address 3001:1::1/64
SwitchA(config-if-VLAN 200)#exit
```

Step 2, Create an OSPFv3 process and specify the router-id

```
SwitchA(config)#ipv6 router ospf  10
SwitchA(config-router)#router-id 1.1.1.1
Change router-id and update OSPFv3 process! [yes/no]:y
SwitchA(config-router)#exit
```

Step 3, Enable OSPFv3 on interface vlan 100, with the area being Area0

```
SwitchA(config)#interface vlan 100
SwitchA(config-if-VLAN 100)#ipv6 ospf 10 area 0
SwitchA(config-if-VLAN 100)#exit
```

Step 4, Enable the OSPFv3 on interface vlan 200, with the area being Area1

```
SwitchA(config)#interface vlan 200
SwitchA(config-if-VLAN 200)#ipv6 ospf 10 area 1
SwitchA(config-if-VLAN 200)#end
```

Configuring SwitchB

Step 1, create a VLAN and configure the IPv6 address

SwitchB# conf

Enter configuration commands, one per line.   End with CNTL/Z.

! Create and configure interface vlan100

```
SwitchB(config)# vlan 100
SwitchB(config-vlan)#interface vlan 100
SwitchB(config-if-VLAN 100)#ipv6 enable
SwitchB(config-if-VLAN 100)#ipv6 address 3001::2/64
SwitchB(config-if-VLAN 100)#exit
```

Step 2, Create an OSPFv3 process and specify the router-id

```
SwitchB(config)#ipv6 router ospf 10
SwitchB(config-router)#router-id 2.2.2.2
Change router-id and update OSPFv3 process! [yes/no]:y
SwitchB(config-router)#exit
```

Step 3, Enable the OSPFv3 on interface vlan 100, with the area being Area0

```
SwitchB(config)#interface vlan 100
SwitchB(config-if-VLAN 100)#ipv6 ospf 10 area 0
SwitchB(config-if-VLAN 100)#end
```

Configuring SwitchC

Step 1, create a VLAN and configure the IPv6 address

SwitchC#configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

! Create and configure interface vlan200

```
SwitchC(config)#vlan 200
SwitchC(config-vlan)#interface vlan 200
SwitchC(config-if-VLAN 200)#ipv6 enable
SwitchC(config-if-VLAN 200)#ipv6 address 3001:1::2/64
SwitchC(config-if-VLAN 200)#exit
```

Step 2, Create an OSPFv3 process and specify the router-id

```
SwitchC(config)#ipv6 router ospf 10
SwitchC(config-router)#router-id 3.3.3.3
Change router-id and update OSPFv3 process! [yes/no]:y
SwitchC(config-router)#exit
```

Step 3, Enable the OSPFv3 on interface vlan 200, with the area being Area1

```
SwitchC (config)#interface vlan 200
SwitchC (config-if-VLAN 200)#ipv6 ospf 10 area 1
SwitchC (config-if-VLAN 200)#end
```

Verifying configuration

Step 1: Verify whether the configuration are correct. Pay attention: whether the router-id is specified, whether the OSPFv3 is enabled on the interface, and whether such parameters as OSPFv3 timer are identical in the same area.

Configuring SwitchA

```
vlan 100
!
vlan 200
!
```

```
interface VLAN 100
 no ip proxy-arp
 ipv6 address 3001::1/64
 ipv6 enable
 ipv6 ospf 10 area 0
!
interface VLAN 200
 no ip proxy-arp
 ipv6 address 3001:1::1/64
 ipv6 enable
 ipv6 ospf 10 area 1
!
ipv6 router ospf 10
 router-id 1.1.1.1
```

Configuring SwitchB

```
vlan 100
!
interface VLAN 100
 no ip proxy-arp
 ipv6 address 3001::2/64
 ipv6 enable
 ipv6 ospf 10 area 0
!
ipv6 router ospf 10
 router-id 2.2.2.2
```

Configuring SwitchC:

```
vlan 200
!
interface VLAN 200
 no ip proxy-arp
 ipv6 address 3001:1::2/64
 ipv6 enable
 ipv6 ospf 10 area 1
!
ipv6 router ospf 10
 router-id 3.3.3.3
```

Step 2: Display OSPFv3 neighbors. Pay attention: whether the adjacencies have been created.

```
SwitchA#show ipv6 ospf neighbor
OSPFv3 Process (10), 2 Neighbors, 2 is Full:
Neighbor ID   Pri  State      Dead Time   Instance ID   Interface
2.2.2.2       1   Full/BDR    00:00:37    0            VLAN 100
3.3.3.3       1   Full/DR     00:00:34    0            VLAN 200
```

The information displayed on SwitchB and SwitchC is similar to the information displayed on SwitchA.

Step 3: Display OSPFv3 routes and ping IPv6 address in another area. Pay attention: whether all the IPv6 routes are learned and whether the routes can be pinged.

```
SwitchC#show ipv6 route
IPv6 routing table name is Default(0) global scope - 7 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
      I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
      O - OSPF intra area, OI - OSPF inter area,  OE1 - OSPF external type 1, OE2 - OSPF external
type 2
      ON1 - OSPF NSSA external type 1, ON2 - OSPF NSSA external type 2
      [*] - NOT in hardware forwarding table
L     ::1/128  via Loopback, local host
OI    3001::/64  [110/2] via FE80::21A:A9FF:FE15:4CB9, VLAN 200
C     3001:1::/64  via VLAN 200, directly connected
L     3001:1::2/128  via VLAN 200, local host
L     FE80::/10  via ::1, Null0
C     FE80::/64  via VLAN 200, directly connected
L     FE80::21A:A9FF:FE01:FB1F/128  via VLAN 200, local host
```

```
SwitchC#ping ipv6 3001::2
Sending 5, 100-byte ICMP Echoes to 3001::2, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/10 ms
```

The information displayed on SwitchA and SwitchB is similar to the information displayed on SwitchC.

OSPFv3 Redistribution Configuration Example

Configuration Requirements

There are three devices which are connected as shown in Figure 2.

- Enable the OSPFv3 protocol on RouterA; Enable BGP protocol and configure the static route on RouterC; For RouterB, redistribute the static route redistributed on RouterC to the OSPFv3 domain. Set the specified community attributes for the static route redistributed to BGP on RouterC and redistribute BGP route with the specified community attributes to the OSPFv3 domain on RouterB.

- Configure the external route summary on RouterB: aggregate the routes within the range of 2001:db8:77::/48 and advertise the summary to the OSPFv3 domain.

- To speed up the convergence, set the SPF calculation delay, holdtime and max-waittime for RouterA and RouterB to 5ms, 1000ms and 90000ms respectively.

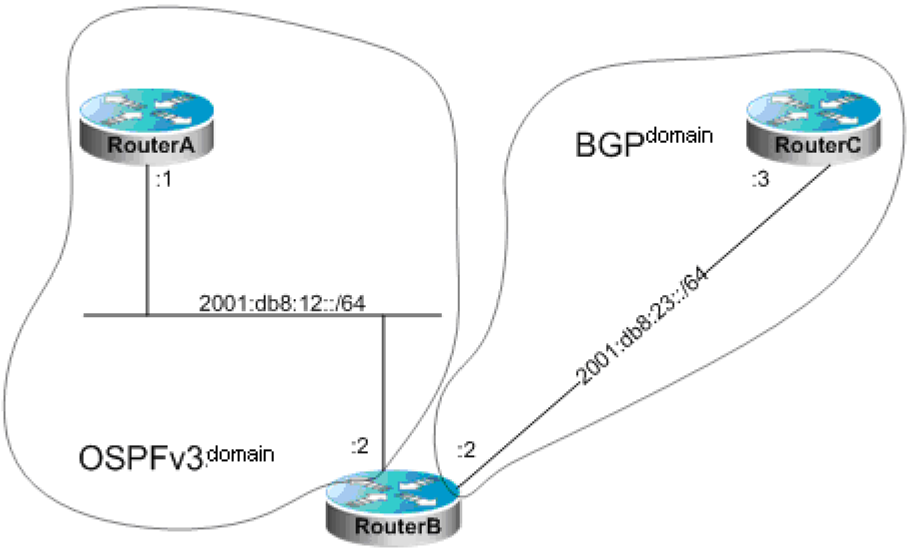


Figure 2 OSPFv3 Redistribution Configuration Example

Configuration Details

Router A configuration:

# Configure the network interface

```
Ruijie(config)# interface gigabitEthernet 0/1
Ruijie(config-if)# ipv6 enable
Ruijie(config-if)# ipv6 address 2001:db8:12::1/64
Ruijie(config-if)# ipv6 ospf 12 area 0
```

# Configure OSPFv3

```
Ruijie(config)# ipv6 router ospf 12
Ruijie(config-router)# router-id 1.1.1.1
Ruijie(config-router)# timers throttle spf 5 1000 90000
```

Router B Configuration:

# Configure the network interface

```
Ruijie(config)# interface gigabitEthernet 0/1
Ruijie(config-if)# ipv6 enable
Ruijie(config-if)# ipv6 address 2001:db8:12::2/64
Ruijie(config-if)# ipv6 ospf 12 area 0
Ruijie(config)# interface gigabitEthernet 0/2
Ruijie(config-if)# ipv6 enable
Ruijie(config-if)# ipv6 address 2001:db8:23::2/64
```

# Configure OSPFv3

```
Ruijie(config)# ipv6 router ospf 12
Ruijie(config-router)# router-id 2.2.2.2
Ruijie(config-router)# redistribute bgp route-map ospfrm
Ruijie(config-router)# timers throttle spf 5 1000 90000
Ruijie(config-router)# summary-prefix 2001:db8:77::/48
```

# Configure BGP

```
Ruijie(config)# router bgp 2
Ruijie(config-router)# neighbor 2001:db8:23::3 remote-as 3
Ruijie(config-router)# address-family ipv6
Ruijie(config-router-af)# neighbor 2001:db8:23::3 activate
```

# Configure route-map

```
Ruijie(config)# route-map ospfrm
Ruijie(config-route-map)# match community cl_110
```

# Define community list

```
Ruijie(config)# ip community-list standard cl_110 permit 22:22
```

Router C Configuration:

# Configure the network interface

```
Ruijie(config)# interface gigabitEthernet 0/1
Ruijie(config-if)# ipv6 enable
Ruijie(config-if)# ipv6 address 2001:db8:23::3/64
```

# Configure BGP

```
Ruijie(config)# router bgp 3
Ruijie(config-router)# neighbor 2001:db8:23::2 remote-as 2
Ruijie(config-router)# address-family ipv6
Ruijie(config-router-af)# redistribute static route-map bgprm
Ruijie(config-router-af)# neighbor 2001:db8:23::2 activate
Ruijie(config-router-af)# neighbor 2001:db8:23::2 send-community
```

# Configure static route

```
Ruijie(config)# ipv6 route 2001:db8:77:88::/64 null 0
Ruijie(config)# ipv6 route 2001:db8:77:99::/64 null 0
```

# Configure route-map

```
Ruijie(config)# route-map bgprm
Ruijie(config-route-map)# set community 22:22
```

Example of Stub Area Configuration

Typology Diagram

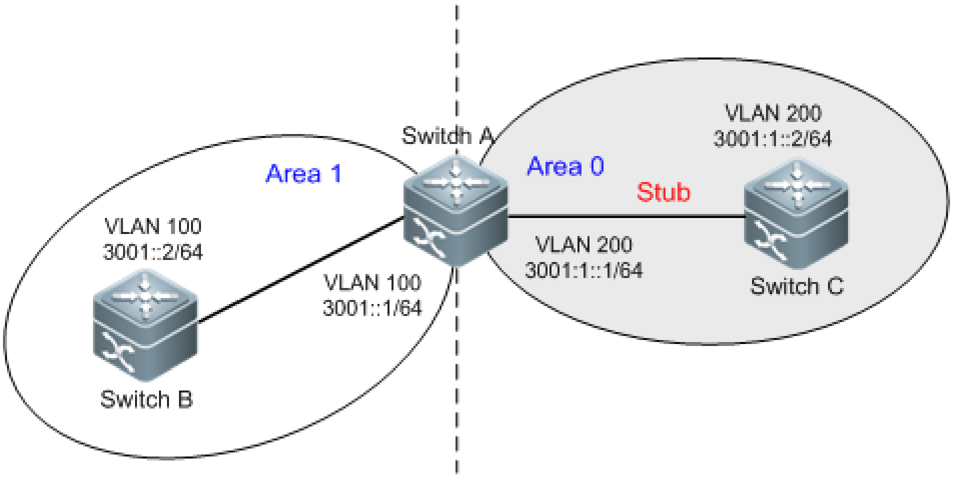


Figure 3 OSPFv3 stub area (the same as Figure 1)

Application Requirements

Configure Area 1 as a stub area in order to reduce the system overhead of switches in this area.

Configuration tips

Use the parameter of "stub no-summary" on the area border router (ABR) (Switch A in this example)

Use the parameter of "stub" on the non-area-border router (Switch C in this example)

Configuration Steps

Configuring SwitchA

Step 1: Enable OSPFv3 basic configuration, as in the OSPFv3 basic configuration example.

Step 2: Configure stub no-summary

```
SwitchA# conf
Enter configuration commands, one per line.  End with CNTL/Z.
SwitchA(config)#ipv6 router ospf 10
SwitchA(config-router)#area 1 stub no-summary
SwitchA(config-router)#exit
```

Configuring SwitchC

Step 1: Enable OSPFv3 basic configuration, as in the OSPFv3 basic configuration example.

Step 2: Configure stub

```
SwitchC# conf
Enter configuration commands, one per line.  End with CNTL/Z.
SwitchA(config)#ipv6 router ospf 10
SwitchA(config-router)#area 1 stub
SwitchA(config-router)#exit
```

Verifying configuration

Step 1: Verify whether the configuration are correct. While making sure the OSPFv3 basic configuration are correct, pay attention to the differences in stub parameters between ABR and the other router.

SwitchA Configuration

```
vlan 100
!
vlan 200
!
interface VLAN 100
 no ip proxy-arp
 ipv6 address 3001::1/64
 ipv6 enable
 ipv6 ospf 10 area 0
!
interface VLAN 200
 no ip proxy-arp
 ipv6 address 3001:1::1/64
 ipv6 enable
 ipv6 ospf 10 area 1
!
ipv6 router ospf 10
 router-id 1.1.1.1
area 1 stub no-summary
!
```

SwitchC Configuration

```
vlan 200
!
interface VLAN 200
 no ip proxy-arp
 ipv6 address 3001:1::2/64
 ipv6 enable
 ipv6 ospf 10 area 1
!
ipv6 router ospf 10
```

```
 router-id 3.3.3.3
area 1 stub
!
```

Step 2: Display OSPFv3 neighbors. Pay attention: whether the adjacencies have been created.

```
SwitchA#show ipv6 ospf neighbor
OSPFv3 Process (10), 2 Neighbors, 2 is Full:
Neighbor ID   Pri  State       Dead Time   Instance ID   Interface
2.2.2.2       1    Full/BDR    00:00:37    0             VLAN 100
3.3.3.3       1    Full/DR     00:00:34    0             VLAN 200
```

Similar information will be displayed on SwitchC.

Step 3: Display OSPFv3 routes. Pay attention: whether the default route is generated, and whether the inter-area route exists

```
SwitchC #show ipv6 route
IPv6 routing table name is Default(0) global scope - 7 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
  I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
  O - OSPF intra area, OI - OSPF inter area,  OE1 - OSPF external type 1, OE2 - OSPF external
type 2
ON1 - OSPF NSSA external type 1, ON2 - OSPF NSSA external type 2
      [*] - NOT in hardware forwarding table
OI    ::/0  [110/2] via FE80::21A:A9FF:FE15:4CB9, VLAN 200
L     ::1/128  via Loopback, local host
C     3001:1::/64  via VLAN 200, directly connected
L     3001:1::2/128  via VLAN 200, local host
L     FE80::/10  via ::1, Null0
C     FE80::/64  via VLAN 200, directly connected
L     FE80::21A:A9FF:FE01:FB1F/128  via VLAN 200, local host
```

Example of OSPFv3 DR Election Configuration

Typology Diagram

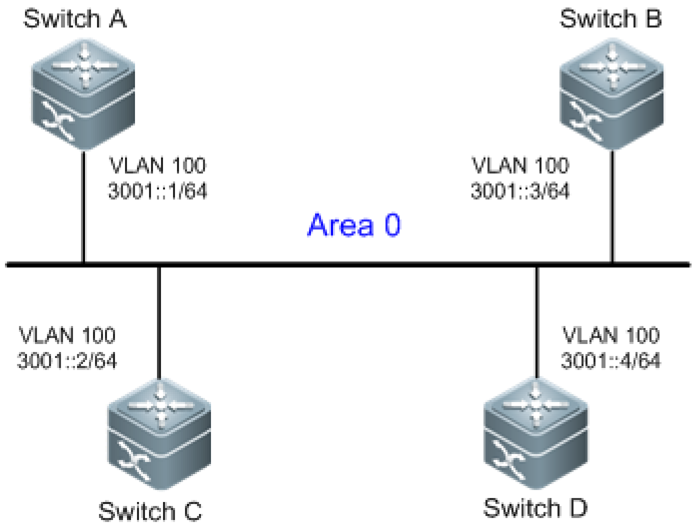


Figure 4 OSPFv3 DR election

SwitchA, SwitchB, SwitchC and SwitchD are in the same area (Area 0) and are interconnected via vlan 100. Switch A and Switch B are devices with the best configuration and the highest stability on the network

Application Requirements

Corresponding requirements: By adjusting the parameter **priority**, configure SwitchA as the DR and SwitchB as the BDR in order to avoid network route flap.

Configuration Tips

Configuration tips

Configure the priority of the interface of expected DR (Switch A in this example) to the highest (150 in this example) and the priority of the interface of BDR (Switch B) to the second highest (50 in this example)

Cautions

The default priority of interface is 1, and DR/BDR can be determined according to the router-id. Generally, the router with the largest router-id will be the DR, and the router with the second largest router-id will be the BDR.

Configuration Steps

Configuring SwitchA

Step 1, create a VLAN and configure the IPv6 address

```
SwitchA# conf

Enter configuration commands, one per line.   End with CNTL/Z.
```

! Create and configure interface vlan100

```
SwitchA(config)# vlan 100
SwitchA(config-vlan)#exit
SwitchA(config)#interface vlan 100
SwitchA(config-if-VLAN 100)#ipv6 enable
SwitchA(config-if-VLAN 100)#ipv6 address 3001::1/64
SwitchA(config-if-VLAN 100)#exit
```

Step 2, Create an OSPFv3 process and specify the router-id

```
SwitchA(config)#ipv6 router ospf 10
SwitchA(config-router)#router-id 1.1.1.1
SwitchA(config-router)#exit
```

Step 3, Enable the OSPFv3 on interface vlan 100, with the area being Area0 and the priority being 150

```
SwitchA(config)#interface vlan 100
SwitchA(config-if-VLAN 100)#ipv6 ospf 10 area 0
SwitchA(config-if-VLAN 100)# ipv6 ospf priority 150
SwitchA(config-if-VLAN 100)#end
```

Configuring SwitchB

Step 1, create a VLAN and configure the IPv6 address

SwitchB# conf

Enter configuration commands, one per line.   End with CNTL/Z.

! Create and configure interface vlan100

```
SwitchB(config)# vlan 100
SwitchB(config-vlan)#exit
SwitchB(config)#interface vlan 100
SwitchB(config-if-VLAN 100)#ipv6 enable
SwitchB(config-if-VLAN 100)#ipv6 address 3001::2/64
SwitchB(config-if-VLAN 100)#exit
```

! Create an OSPFv3 process and specify the router-id

```
SwitchB(config)#ipv6 router ospf 10
SwitchB(config-router)#router-id 2.2.2.2
SwitchB(config-router)#exit
```

Step 2, Enable the OSPFv3 on interface vlan 100, with the area being Area0 and the priority being 50

```
SwitchB(config)#interface vlan 100
SwitchB(config-if-VLAN 100)#ipv6 ospf 10 area 0
SwitchB(config-if-VLAN 100)# ipv6 ospf priority 50
SwitchB(config-if-VLAN 100)#end
```

Configuring SwitchC

Step 1, create a VLAN and configure the IPv6 address

SwitchC# conf

Enter configuration commands, one per line.   End with CNTL/Z.

! Create and configure interface vlan100

```
SwitchC(config)#vlan 100
SwitchC(config-vlan)#exit
SwitchC(config)##interface vlan 100
SwitchC(config-if-VLAN 100)#ipv6 enable
SwitchC(config-if-VLAN 100)#ipv6 address 3001::3/64
SwitchC(config-if-VLAN 100)#exit
```

Step 2, Create an OSPFv3 process and specify the router-id

```
SwitchC(config)#ipv6 router ospf 10
SwitchC(config-router)#router-id 3.3.3.3
SwitchC(config-router)#exit
```

Step 3, Enable OSPFv3 on interface vlan 100, with the area being Area0 and the priority using the default value.

```
SwitchC(config)#interface vlan 100
SwitchC(config-if-VLAN 100)#ipv6 ospf 10 area 0
SwitchC(config-if-VLAN 100)#end
```

Configuring SwitchD

Step 1, create a VLAN and configure the IPv6 address

SwitchD# conf

Enter configuration commands, one per line. End with CNTL/Z.

! Create and configure interface vlan100

```
SwitchD(config-vlan)#vlan 100
SwitchD(config-vlan)#exit
SwitchD(config)#interface vlan 100
SwitchD(config-if-VLAN 100)#ipv6 enable
SwitchD(config-if-VLAN 100)#ipv6 address 3001::4/64
SwitchD(config-if-VLAN 100)#exit
```

Step 2, Create an OSPFv3 process and specify the router-id

```
SwitchD(config)#ipv6 router ospf 10
SwitchD(config-router)#router-id 4.4.4.4
SwitchD(config-router)#exit
```

Step 3, Enable OSPFv3 on interface vlan 100, with the area being Area0 and the priority using the default value.

```
SwitchD(config)#interface vlan 100
SwitchD(config-if-VLAN 100)#ipv6 ospf 10 area 0
SwitchD(config-router)#end
```

Verifying configuration

Step 1: Verify whether the configuration are correct. Pay attention: whether the OSPF basic parameters and interface priority are correct

SwitchA Configuration:

```
vlan 100
!
interface VLAN 100
 no ip proxy-arp
 ipv6 address 3001::1/64
 ipv6 enable
 ipv6 ospf 10 area 0
ipv6 ospf priority 150
!
ipv6 router ospf 10
 router-id 1.1.1.1
```

SwitchB Configuration

```
vlan 100
!
interface VLAN 100
 no ip proxy-arp
 ipv6 address 3001::2/64
 ipv6 enable
 ipv6 ospf 10 area 0
ipv6 ospf priority 50
!
ipv6 router ospf 10
 router-id 2.2.2.2
```

SwitchC Configuration

```
vlan 100
!
interface VLAN 100
 no ip proxy-arp
 ipv6 address 3001::3/64
 ipv6 enable
 ipv6 ospf 10 area 0
!
ipv6 router ospf 10
 router-id 3.3.3.3
```

SwitchD Configuration:

```
vlan 100
!
interface VLAN 100
 no ip proxy-arp
 ipv6 address 3001::4/64
 ipv6 enable
 ipv6 ospf 10 area 0
!
ipv6 router ospf 10
 router-id 4.4.4.4
```

Step 2: Display OSPFv3 neighbors. Pay attention: whether the adjacencies have been created, and whether each switch plays the correct role.

```
SwitchD#show ipv6 ospf neighbor
Neighbor ID  Pri   State    Dead Time   Interface ID    Interface
3.3.3.3      1   2WAY/DROTHER  00:00:33    4196         Vlan100
1.1.1.1      150   FULL/DR     00:00:35    4196           Vlan100
2.2.2.2      50   FULL/BDR     00:00:35    4196           Vlan100
```

Adjacencies before priority configuration are shown below. We can see that DR/BDR can be specified by adjusting the priority.

```
SwitchA#show ipv6 ospf neighbor
OSPFv3 Process (10), 3 Neighbors, 2 is Full:
Neighbor ID  Pri   Stat      Dead Time   Instance ID   Interface
2.2.2.2       1    Full/BDR       00:00:33    0            VLAN 100
3.3.3.3       1    2-Way/DROther   00:00:35   0            VLAN 100
4.4.4.4       1    Full/DR        00:00:33    0            VLAN 100
```

Configuration Example of OSPFv3 Multiple Instances on One Link

Topological Diagram

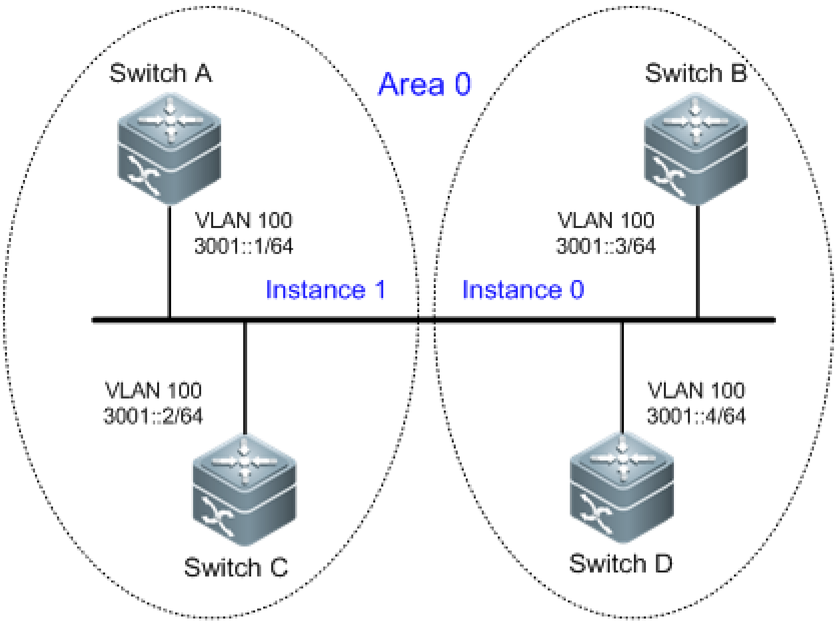


Figure 1-5 Multiple instances on one link

SwitchA, SwitchB, SwitchC and SwitchD are in the same area (Area 0) and are interconnected via vlan 100.

Application Requirements

On a broadcast link, especially within a vlan, adjacencies will be established among all the switches. This may result in increased system overhead and network oscillation.

Application requirements: Switches in the same area are divided into several groups, and OSPFv3 adjacencies can only be established between switches belonging to the same group.

Configuration Tips

Key points

By configuring multiple instances on the same link (the link on interface vlan100 in this example), adjacency establishment by group can be implemented (in this example, SwitchA and SwitchB form a group, with instance ID being 1; SwitchC and SwitchD form a group, with instance ID being 0).

Cautions

By default, the instance ID of the interface is 0. In this example, you only need to configure the instance ID on Switch A and Switch B.

Configuration Steps

Configuring SwitchA

Step 1, Create a VLAN and configure the IPv6 address

SwitchA# conf

Enter configuration commands, one per line. End with CNTL/Z.

! Create and configure interface vlan100

```
SwitchA(config)# vlan 100
SwitchA(config-vlan)#interface vlan 100
SwitchA(config-if-VLAN 100)#ipv6 enable
SwitchA(config-if-VLAN 100)#ipv6 address 3001::1/64
SwitchA(config-if-VLAN 100)#exit
```

Step 2, Create an OSPFv3 process and specify the router-id

```
SwitchA(config)#ipv6 router ospf 10
SwitchA(config-router)#router-id 1.1.1.1
SwitchA(config-router)#exit
```

Step 3, Enable the OSPFv3 on interface vlan 100, with the area being Area0 and the instance ID being 1

```
SwitchA(config)#interface vlan 100
SwitchA(config-if-VLAN 100)#ipv6 ospf 10 area 0 instance 1
SwitchA(config-if-VLAN 100)# end
```

Configuring SwitchB

Step 1, Create a VLAN and configure the IPv6 address

SwitchB# conf

Enter configuration commands, one per line. End with CNTL/Z.

! Create and configure interface vlan100

```
SwitchB(config)# vlan 100
SwitchB(config-vlan)#interface vlan 100
SwitchB(config-if-VLAN 100)#ipv6 enable
SwitchB(config-if-VLAN 100)#ipv6 address 3001::2/64
SwitchB(config-if-VLAN 100)#exit
```

Step 2, Create an OSPFv3 process and specify the router-id

```
SwitchB(config)#ipv6 router ospf 10
SwitchB(config-router)#router-id 2.2.2.2
SwitchB(config-router)#exit
```

Step 3, Enable the OSPFv3 on interface vlan 100, with the area being Area0 and the instance ID being 1

```
SwitchB(config)#interface vlan 100
SwitchB(config-if-VLAN 100)#ipv6 ospf 10 area 0 instance 1
SwitchB(config-if-VLAN 100)# end
```

Verifying Configuration

Step 1: Verify whether the configuration is correct. Pay attention: whether the instance ID for establishing adjacency between switches is correct.

SwitchA Configuration:

```
vlan 100
!
interface VLAN 100
 no ip proxy-arp
 ipv6 address 3001::1/64
 ipv6 enable
 ipv6 ospf 10 area 0 instance 1
!
ipv6 router ospf 10
 router-id 1.1.1.1
```

SwitchB Configuration:

```
vlan 100
!
interface VLAN 100
 no ip proxy-arp
 ipv6 address 3001::2/64
 ipv6 enable
ipv6 ospf 10 area 0 instance 1
!
ipv6 router ospf 10
 router-id 2.2.2.2
```

SwitchC Configuration:

```
vlan 100
!
interface VLAN 100
 no ip proxy-arp
 ipv6 address 3001::3/64
 ipv6 enable
 ipv6 ospf 10 area 0
!
```

```
ipv6 router ospf 10
 router-id 3.3.3.3
```

SwitchD Configuration:

```
vlan 100
!
interface VLAN 100
 no ip proxy-arp
 ipv6 address 3001::4/64
 ipv6 enable
 ipv6 ospf 10 area 0
!
ipv6 router ospf 10
 router-id 4.4.4.4
```

Step 2: Display the instance ID of the interface link and reconfirm that the switches in the same group have the same instance ID

```
SwitchA#show ipv6 ospf interface vlan 100
VLAN 100 is up, line protocol is up
  Interface ID 4196
  IPv6 Prefixes
    fe80::21a:a9ff:fe15:4cb9/64 (Link-Local Address)
    3001::1/64
  OSPFv3 Process (10), Area 0.0.0.0, Instance ID 1
    Router ID 1.1.1.1, Network Type BROADCAST, Cost: 1
    Transmit Delay is 1 sec, State BDR, Priority 1
    Designated Router (ID) 2.2.2.2
      Interface Address fe80::2d0:f8ff:fe22:88b1
    Backup Designated Router (ID) 1.1.1.1
      Interface Address fe80::21a:a9ff:fe15:4cb9
    Timer interval configured, Hello 10, Dead 40, Wait 40, Retransmit 5
      Hello due in 00:00:08
    Neighbor Count is 1, Adjacent neighbor count is 1
  Hello received 7 sent 8, DD received 3 sent 5
  LS-Req received 1 sent 1, LS-Upd received 5 sent 4
  LS-Ack received 3 sent 3, Discarded 0
```

```
SwitchB#show ipv6 ospf interface vlan 100
VLAN 100 is up, line protocol is up
  Interface ID 4196
  IPv6 Prefixes
    fe80::2d0:f8ff:fe22:88b1/64 (Link-Local Address)
    3001::2/64
  OSPFv3 Process (10), Area 0.0.0.0, Instance ID 1
    Router ID 2.2.2.2, Network Type BROADCAST, Cost: 1
```

```
   Transmit Delay is 1 sec, State DR, Priority 1
   Designated Router (ID) 2.2.2.2
     Interface Address fe80::2d0:f8ff:fe22:88b1
   Backup Designated Router (ID) 1.1.1.1
     Interface Address fe80::21a:a9ff:fe15:4cb9
   Timer interval configured, Hello 10, Dead 40, Wait 40, Retransmit 5
     Hello due in 00:00:05
   Neighbor Count is 1, Adjacent neighbor count is 1
  Hello received 16 sent 21, DD received 10 sent 8
  LS-Req received 2 sent 2, LS-Upd received 10 sent 9
  LS-Ack received 6 sent 6, Discarded 0
```

Step 3: Display OSPFv3 neighbors. Pay attention: whether the adjacencies have been created, and whether the adjacencies are established only between the switches in the same group.

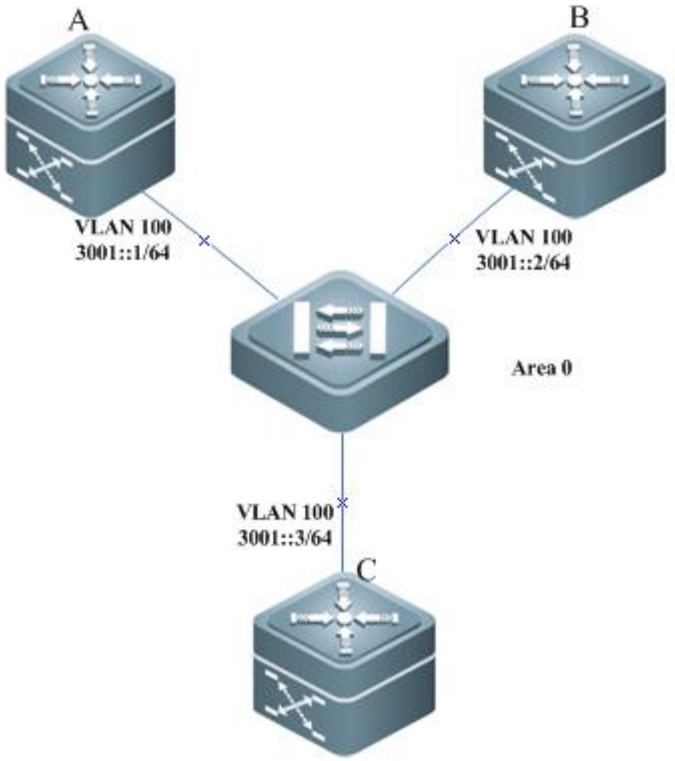
```
SwitchA#show ipv6 ospf neighbor
OSPFv3 Process (10), 1 Neighbors, 1 is Full:
Neighbor ID  Pri State     Dead Time   Instance ID   Interface
2.2.2.2       1   Full/DR   00:00:39     1           VLAN 100
```

```
SwitchB#show ipv6 ospf neighbor
OSPFv3 Process (10), 1 Neighbors, 1 is Full:
Neighbor ID  Pri  State       Dead Time   Instance ID   Interface
1.1.1.1       1   Full/BDR    00:00:34    1            VLAN 100
```

Configuration Example of OSPFv3 Authenticated Encryption

Topological Diagram

Figure 1-6 Encrypted authentication instance



Application Requirements

The OSPFv3 protocol runs on Device A, B and C, which belong to the same area. Connected interfaces establish adjacency only if the same authenticated encryption is configured on them.

The interfaces connecting Device A and B are configured with the same authentication parameters. Device C is not configured with authentication. Device A establishes adjacency with Device B and no adjacency with Device C.

Configuriation Tips

According to the topological requirements, the OSPFv3 protocol is configured to run on Device A, B and C, which belong to the same area

The Interfaces connecting Device A and B are configured with the same authentication parameters.

Configuration Steps

- Configuring Switch A

Step1, Enable the same OSPFv3 basic configuration on Device A, B and C as OSPFv3 basic configuration example.

Step2. Configure authentication parameters on interfaces connecting Device A and B.

```
Ruijie(config)# vlan 100
Ruijie(config-vlan)# exit
Ruijie(config)# interface vlan 100
Ruijie(config-if-vlan 100)# ipv6 address 3001::1/64
Ruijie(config-if-vlan 100)# ipv6 ospf 1 area 0
Ruijie(config-if-vlan 100)# ipv6 ospf authentication ipsec spi 400 md5
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
Ruijie(config-if-vlan 100)# exit
```

- Configuring Switch B

Step1, Enable the same OSPFv3 basic configuration on Device A, B and C as OSPFv3 basic configuration example.

Step2. Configure authentication parameters on interfaces connecting Device A and B.

```
Ruijie(config)# vlan 100
Ruijie(config-vlan)# exit
Ruijie(config)# interface vlan 100
Ruijie(config-if-vlan 100)# ipv6 address 3001::2/64
Ruijie(config-if-vlan 100)# ipv6 ospf 1 area 0
Ruijie(config-if-vlan 100)# ipv6 ospf authentication ipsec spi 400 md5
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
Ruijie(config-if-vlan 100)# exit
```

- Configuring Switch C

Step1, Enable the same OSPFv3 basic configuration on Device A, B and C as OSPFv3 basic configuration example.

Step2. Configure authentication parameters on interfaces connecting Device A and B.

```
Ruijie(config)# vlan 100
Ruijie(config-vlan)# exit
Ruijie(config)# interface vlan 100
Ruijie(config-if-vlan 100)# ipv6 address 3001::2/64
Ruijie(config-if-vlan 100)# ipv6 ospf 1 area 0
Ruijie(config-if-vlan 100)# ipv6 ospf authentication ipsec spi 400 md5
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
Ruijie(config-if-vlan 100)# exit
```

Verifying Configuration

- Step 1: Run the **show ipv6 ospf neighbor** command on Device A, B and C. It is shown that each two devices establish adjacency.
- Step 2: Neighbors.are shown only on Device A and B. Device C shows no neighbor. Run the **debug ipv6 ospf packet** command. Device A and B can receive packets from each other while Device C cannot receive packets.

```
Ruijie#show ipv6 ospf neighbor
OSPFv3 Process (1), 1 Neighbors, 1 is Full:
Neighbor ID   Pri   State      Dead Time   Instance ID   Interface
2.2.2.2      1    Full/BDR   00:00:37    0           VLAN 100

Ruijie#show ipv6 ospf neighbor
OSPFv3 Process (1), 1 Neighbors, 1 is Full:
Neighbor ID   Pri   State      Dead Time   Instance ID   Interface
1.1.1.1      1    Full/BDR   00:00:37    0           VLAN 100

Ruijie#show ipv6 ospf neighbor
```

```
OSPFv3 Process (1), 0 Neighbors, 0 is Full:
```

Configuring BGP

About BGP

The border gateway protocol (BGP) is an exterior gateway protocol (EGP) used for routers to communicate with one another in different autonomous systems. The protocol is designed to exchange information about network reachability among these autonomous systems (AS) and eliminate loops based on the characteristics of the BGP protocol.

The BGP protocol relies on the TCP protocol for reliable packet transmission.

A router which operates on the BGP protocol is referred to as a “BGP Speaker”, and BGP Speakers that have set up a BGP session are referred to as “BGP Peers”.

There are two modes of BGP session: IBGP (Internal BGP) and EBGP (External BGP). The IBGP refers to a BGP session in an AS, while the EBGP refers to a BGP session between different ASs. To summarize, the EBGP exchanges routing information among different ASs. The IBGP transmits routing information in an AS.

The BGP protocol has the following features:

- Supports BGP-4
- Supports path attributes
  - ✓ ORIGN Attribute
  - ✓ AS_PATH Attribute
  - ✓ NEXT_HOP Attribute
  - ✓ MULTI_EXIT_DISC Attribute
  - ✓ LOCAL-PREFERENCE Attribute
  - ✓ ATOMIC_AGGREGATE Attribute
  - ✓ AGGREGATOR Attribute
  - ✓ COMMUNITY Attribute
  - ✓ ORIGINATOR_ID Attribute
  - ✓ CLUSTER_LIST Attribute
  - ✓ AS4_PATH Attribute
  - ✓ AS4_AGGREGATOR Attribute
  - ✓ Connector Attribute
- Supports BGP peer groups
- Supports loopback interface
- Supports MD5 authentication of TCP
- Supports the synchronization of BGP and IGP
- Supports the aggregation of BGP routes
- Supports BGP route flap dampening
- Supports BGP routing reflector
- Supports AS confederation
- Supports BGP soft reset
- Supports BGP Graceful Restart (defined in RFC4724)

Enabling the BGP Protocol

To enable the BGP protocol, execute the following commands in privileged EXEC mode:

Command	**Function**
Ruijie# configure terminal	Enters global configuration mode.
Ruijie(config)# **ip routing**	Enables the routing function (if the switch is disabled).
Ruijie(config)# **router bgp** *as-number*	Enables the BGP and configures the AS number. The range of *AS-number* is 1 to 65535.
Ruijie(config-router)# **bgp router-id** *router-id*	(Optional) Configures the ID used when this switch runs the BGP protocol.
Ruijie(config-router)# **end**	Returns to privileged EXEC mode.
Ruijie# **show run**	Shows current configuration.
Ruijie# copy running-config startup-config	Saves the configuration.

Use the **no router bgp** command to disable the BGP protocol**.**

Default BGP Configuration

The BGP protocol is not enabled by default.

After the BGP protocol is enabled, the default BGP configuration is shown as follows:

Feature		**Default Setting**
Router ID		To configure a loopback interface, select the maximum address from loopback interface addresses. Otherwise, select the maximum interface address from the directly connected interface.
Synchronization of BGP and IGP		Enabled
Generation of Default Route		Disabled
Multi hops of EBGP	Status	Off
	Number of hops	255
TCP MD5 Authentication		Disabled
Timer	Keepalive Time	60 seconds
	Holdtime	180 seconds
	ConnectRetry Time	120 seconds
	AdvInterval(IBGP)	15 seconds
	AdvInterval(EBGP)	30 seconds
Path Attribute	MED	0
	LOCAL_PREF	100
Route Aggregate		Off
Route Flap Dampening	Status	Off
	Suppress Limit	2000
	Half-life-time	15 minutes
	Reuse Limit	750

Feature		Default Setting
	Max-suppress-time	4*half-life-time
Route Reflector	Status	Off
	Cluster ID	Undefined
	Route among reflection clients	Enabled
AS Confederation		Off
Soft Reset		Off
Traceful Restart		Disabled
Management Distance	External-distance	20
	Internal-distance	200
	Local-distance	200

Injecting Routing Information into the BGP Protocol

The BGP protocol has no routing information when running for the first time. There are two ways to inject routing information to the BGP:

Manually inject routing information to the BGP by using the **network** commands.

Inject routing information to the BGP from the IGP protocol through interaction with the IGP protocol.

The BGP will advertise the injected routing information to its neighbors. This section outlines the manual injection of routing information. For injecting routing information from the IGP protocol, refer to the *Configuration of BGP and IGP Interaction* in related sections.

To manually inject network information advertised by the BGP Speaker to other BGP Speaker, execute the following commands in BGP configuration mode:

Command	Function
Router(config-router)# **network** *network-number* **mask** *network-mask*[**route-map** *map-tag*]	Configures the network whose routing information will be injected into the BGP routing table.

Use the **no network** *network-number* **mask** *network-mask* command to remove the configuration. To cancel the used route-map, reconfigure it by using the r*oute-map not added* option. If the configured network information comes under standard class A, class B or class C network address, the mask option of this command cannot be used.

In BGP4+, you can use this command in IPv6 address family configuration mode to configure IPv6 routes.

Caution

- The **network** command is used to inject IGP routes into the routing table of BGP, and the advertised networks can be direct-connected, static and dynamic routes.
- For the external gateway protocol (EGP), the **network** command indicates the network to be advertised. This is different from the internal gateway protocol (IGP, such as OSPF and RIP). The IGP uses the **network** commands to determine where the routing update message will be sent to.

Sometimes, you may need to use an IGP route rather than an EBGP route. This can be done by using the **network backdoor** command. Execute the following operations in BGP configuration mode:

Command	Function
Ruijie(config-router)# **network** *network-number* **mask** *network-mask* **backdoor**	Sets the backdoor route.

Use the **no network** *network-number* **mask** *network-mask* **backdoor** command to remove the configuration.

Caution

By default, the distance for network information management learned from the BGP Speakers which have established the EBGP connection is 20. Set the distance by using the **network backdoor** command as 200. As such, identical network information learned from the IGP presents a higher priority. The networks learned from the IGP are considered as backdoor networks, and will not be advertised.

Controlling Route Advertisement

The BGP protocol can control the routes advertised to the core routing table by using the **table-map** command. If a route is matched, the command modifies its attribute and advertises it. If a route is not matched or denied, the command advertises it without modifying its attribute.

By default, the **table-map** command advertises all routes without modifying their attributes.

To configure the **table-map** command, execute it in BGP configuration mode or IPv4 address family confiugration mode:

Command	Function
Router(config-router)# **table-map** *route-map-name*	Configures table-map. route-map-name indicates the name of the route-map you want to associate.

Use the **no table-map** command to remove the configuration.

For the configuration of the **table-map** command to take effect immediately, run the **clear ip bgp** [vrf *vrf-name*] **table-map** command to update the core routing table. The **clear ip bgp** [vrf *vrf-name*] **table-map** command will clear but add the routes in the core routing table. Instead, it uses the table-map to advertise route update messages without causing forwarding oscillation..

The **table-map** command supports the following options: rules-match, as-path/community/ip address/ip next-hop/metric/origin/route-type, set, metric/tag/next-hop.

Controlling the Route Redistribution from IBGP to IGP

The BGP protocol controls the redistribution of routes learnded from the IBGP protocol to IGP protocol by using the **bgp redistribute-internal** command. The routes learned from the EBGP protocol or confederation can be redistriubted to the IGP protocol.

This command is enabled by default in either VRF or global mode. Specifically, routers learned from the IBGP can be redistributed to the IGP protocol.

To redistribute a route to the IGP protocol (inclding RIP/OSPF/ISIS), execute the following command in BGP configuration mode, IPv4/IPv6 address family configuration mode or IPv4 VRF address family configuraiton mode:

Command	Function
Router(config-router)# **bgp-redistirbute-internal**	Redistributes IBGP routes to the IGP protocol.

Use the **no bgp redistribute-internal** command to remove the configuration.

Configuring BGP Peer (Group) and Its Parameters

Since the BGP is an external gateway protocol (EGP), it is necessary for a BGP Speaker to know who its peer (BGP Peer) is.

As mentioned in the overview of the BGP protocol, two modes can be used to set up the connection among BGP Speakers: IBGP (Internal BGP) and EBGP (External BGP). The protocol determines which connection will be established among BGP Speakers by using the AS of BGP Peer and BGP Speakers.

The BGP protocol supports IPv4 and IPv6. To check IPv6 function, verify whether the **address-family ipv6** command is executed in BGP configuration mode. Otherwise, IPv6 is not supported. An IPv4 address represents an IPv4 neighbor. An IPv6 address represents an IPv6 neighbor. Note that you should activate neighbors in the right address family.

In general, BGP Speakers with EBGP connection should be physically connected. BGP Speakers with IBGP connection, however, can be located anywhere within an AS.

To configure the BGP peer, execute the following commands in BGP configuration mode:

Command	Function
Ruijie(config-router)# **neighbor** {*address* | *peer-group-nam*e } **remote-as** *as-number*	Configures the BGP peer. *address* indicates the IP addresses of the BGP peer. *peer-group-name* indicates the name of the BGP peer group.

Command	Function
	The range of *as-number* is 1 to 65535.

Use the **no neighbor** {*address*|*peer-group-name*} to delete one peer or peer group.

The BGP Speakers have some configurations in common (including the executed routing policy). To simplify configuration and improve efficiency, it is recommended that you use the BGP peer group.

To configure the BGP peer group, execute the following commands in BGP configuration mode:

Command	Function
Ruijie(config-router)# **neighbor** *peer-group-name* **peer-group**	Creates a BGP peer group.
Ruijie(config-router)# **neighbor** *peer-group-name* **remote-as** *as-number*	(Optional) Configures the BGP peer group. The range of *as-number* is 1 to 4294967295.
Ruijie(config-router)# **neighbor** *address* **peer-group** *peer-group-name*	(Optional) Sets the BGP peer as the member of the BGP peer group.

Use the **no neighbor** *address* **peer-group** to delete some members of the BGP peer group.

Use the **no neighbor** *peer-group-name* **peer-group** to delete the entire peer group.

Use the **no neighbor** *peer-group-name* **remote-as** to delete all members of the BGP peer group and AS numbers of the peer group.

To configure the peer of the BGP Speakers or the optional parameter of the BGP peer group, execute the following commands in BGP configuration mode:

Command	Function
Ruijie(config-router-af)# **neighbor** {*address* | *peer-group-name*} **activate**	(Optional) Activates the address family of the neighbor so that the router can exchange routing information with the address family.
Ruijie(config-router)# **neighbor** {*address* | *peer-group-name*} **update-source** *interface*	(Optional) Configures the network interfaces to establish the BGP session with specified BGP peer (group).
Ruijie(config-router)# **neighbor** {*address* | *peer-group-name*} **ebgp-multihop** [*ttl*]	(Optional) Allows you to establish a BGP session among non-direct-connected EBGP peers (group). The range of TTL is 1 to 255, the EBGP is one hop by default, and the IBGP is 255 hops by default.
Ruijie(config-router)# **neighbor** {*address* | *peer-group-name*} **password** *string*	(Optional) Enables the TCP MD5 authentication when the connection is established among specified BGP peer (group), and configures the password.
Ruijie(config-router)# **neighbor** {*address* | *peer-group-name*} **times** *keepalive holdtime*	(Optional) Configures the Keepalive and Holdtime value to establish a connection with the specified BGP peer (group). The range of the *keepalive* is 0 to 65535 seconds, 60 seconds by default. The range of the *holdtime* is 0 to 65535 seconds, 180 seconds by default.
Ruijie(config-router)# **neighbor** {*address* | *peer-group-name*} **advertisemet-interval** *seconds*	(Optional) Configures the minimal time interval of sending the routing update message to the specified BGP peer (group). The range of advertisement-interval is 1 to 600 seconds, 15

Command	Function
	seconds for the IBGP peer by default, and 30 seconds for the EBGP peer by default.
Ruijie(config-router)# **neighbor** {*address* | *peer-group-name*} **default-originate** [**route-map** *map-tag*]	(Optional) Configures the router to send a default route to the specified BGP peer (group).
Ruijie(config-router)# **neighbor** {*address* | *peer-group-name*} **next-hop-self**	(Optional) Configures the router to set the next routing information as this BGP speaker when the route is distributed to the specified BGP peer (group).
Ruijie(config-router)# **neighbor** {*address* | *peer-group-name*} **remove-private-as**	(Optional) Configures the router to delete the private AS number in the AS path attribute when distributing the routing information to the EBGP peer (group).
Ruijie(config-router)# **neighbor** {*address* | *peer-group-name*} **send-community**	(Optional) Configures the router to send the community attribute to the specified BGP peer (group).
Ruijie(config-router)# **neighbor** {*address* | *peer-group-name*} **maximum-prefix** *maximum* [**warning-only**]	(Optional) Limits the number of the messages received from the specified BGP peer (group).
Ruijie(config-router)# **neighbor** {*address* | *peer-group-name*} **distribute-list** *access-list-name* {**in** | **out**}	(Optional) Configures the router to implement the routing police according to the access control list when routing information is received from and sent to the specified BGP peer (group).
Ruijie(config-router)# **neighbor** {*address* | *peer-group-name*} **prefix-list** *prefix-list-name* {**in** | **out**}	(Optional) Configures the router to implement the routing policy according to the prefix list when the routing information is received from and sent to specified BGP peer (group).
Ruijie(config-router)# **neighbor** {*address* | *peer-group-name*} **route-map** *map-tag* {**in** | **out**}	(Optional) Configures the router to implement the routing policy according to the route-map when the routing information is received from and sent to the specified BGP peer (group).
Ruijie(config-router)# **neighbor** {*address* | *peer-group-name*} **filter-list** *path-list-name* {**in** | **out**}	(Optional) Configures the router to implement the routing policy according to the AS path list when the routing information is received from and sent to the specified BGP peer (group).
Ruijie(config-router)# **neighbor** {*address* | *peer-group-name*} **unsuppress-map** *map-tag*	(Optional) Configures the router to selectively advertise the routing information suppressed by the **aggregate-address** command previously when it is distributed to the specified BGP peer.
Ruijie(config-router)# **neighbor** {*address* | *peer-group-name*} **soft-reconfiguration** inbound	(Optional) Restarts the BGP session and reserve the unchanged routing information sent by the BGP peer (group).
Ruijie(config-router)# **neighbor** {*address* | *peer-group-name*} **route-reflector-client**	(Optional) Configures this switch as the route reflector and specify its client.
Ruijie(config-router)# **neighbor** {*address* | *peer-group-name*} **shutdown**	(Optional) Disables the BGP peer (group).

Use the **no** form of the above commands to disable the configurations.

If a peer does not support **remote-as**, each of its members can use the **neighbor remote-as** command to configure it independently.

By default, each member of the BGP peer group will inherit all its configurations. However, each member can support the optional configurations without affecting the output update independently to replace the unified configuration of the BGP peer group.

Caution

Each member of the BGP peer group can support the optional configurations without affecting the output update independently to replace the unified configuration of the BGP peer group. That is to say, each member of the BGP peer group will inherit the following configurations: **remote-as**, **update-source**, **local-as**, **reconnect-interval**, **times**, **advertisemet-interval**, **default-originate**, **next-hop-self**, **password remove-private-as**, **send-community**, **distribute-list out**, **filter-list out**, **prefix-list out**, **route-map out**, **unspress-map**, **route-reflector-client.**

The **neighbor update-source** command can be used to select any valid interface for establishing a TCP connection. This command is designed mostly to provide available Loopback interfaces, which increase the stability of the connection to the IBGP Speaker.

By default, direct physical connection with BGP peers is required for establishing an EBGP connection. To establish the EBGP peers among non-direct-connected external BGP Speakers, the **neighbor ebgp-multihop** command can be used.

Caution

To avoid route loop and oscillation, the EBGP peers who need multiple hops for BGP connection must have non-default routes to each other.

For the sake of security, you can set the authentication for the BGP peers (group) which will establish the connection based on the MD5 algorithm. The authentication password for the BGP peer should be identical. The process of enabling the MD5 authentication on a BGP peer is shown as follows:

Command	Function
Ruijie(config-router)# **neighbor** {*address* | *peer-group-name*} **password** *string*	Enables the TCP MD5 authentication and set the password when the BGP connection with the BGP peer is established.

Use the **no neighbor** {*address* | *peer-group-name*} **password** command to disable the BGP peer (group) from MD5 authentication.

Use the **neighbor shutdown** command to disable the valid connection established with the BGP peer (group), and delete all routing information related to the BGP peer (group).

Caution

To break the connection established with the specified BGP peer (group) and reserve the configuration information set for this specified BGP peer (group), use the **neighbor shutdown** command. If such configuration information is no longer required, use the **no neighbor** [**peer-group**] command.

Configuring the Management Policy

Whenever the routing policy (including the **neighbor distribute-list**, **neighbor route-map**, **neighbor prefix-list and neighbor filter-list)** changes, you need implement the new routing policy. The traditional way is to break and re-establish the BGP session.

This product supports implementing a new routing policy without ending the BGP session by using soft reset for BGP effectively.

To facilitate the description of BGP soft reset, the following section will refer to the routing policy that affects the input routing information as the input routing policy (such as the **In-route-map** and **In-dist-list**) and the policy that affects the output routing information as the output routing policy (such as the **Out-route-map** and **Out-dist-list**).

If the output routing policy changes, execute the following commands in BGP configuration mode:

Command	Function
Ruijie# **clear ip bgp** {* | peer *address* | **peer-group** *peer-group-name* | **external**} **soft out**	Soft-resets the BGP session and executes the routing policy without resetting the BGP session.

A change in input routing policy changes complicates operations compared with the output routing policy, because the output routing policy is based on the routing table of this BGP Speaker. The implementation of the input routing policy is based on the routing information received from a BGP peer. To reduce memory consumption, the local BGP Speaker will not retain the original routing information received from BGP peers.

To modify the input routing policy if necessary, save the original routing information for each specified BGP peer in this BGP Speaker by using the **neighbor soft-reconfiguration inbound** command. The aim is to provide the original foundation of routing information to modify the input routing policy.

At present, the standard implementation method is referred to as the “Route Refresh Performance”, which can support modifying the routing policy without storing the original routing information. This product supports the feature.

If the input routing policy changes, execute the following commands in BGP configuration mode:

Command	Function
Ruijie(config-router)# **neighbor** {*address* | *peer-group-name*} **soft-reconfiguration inbound**	Restarts the BGP session and reserve the unchanged routing information from the BGP peer (group). This command may consume more memory. If both parties support route refreshing performance, this command becomes unnecessary.
Ruijie# clear **ip bgp** {* | *peer-address* | **peer-group** *peer-group-name* | **external**} **soft in**	Soft-resets the BGP session and executes the routing policy without resetting the BGP session.

You can determine whether the BGP peer supports route refreshing performance by the **show ip bgp neighbors** command. If so, you need to execute the **neighbor soft-reconfiguration inbound** command when the input routing policy changes.

Configuring Synchronization between BGP and IGP

The routing information can be transmitted to another AS through the local AS only when it passes through this AS and reaches another AS. The routing information will be advertised to all the routers in the local AS. Otherwise, if some routers running the IGP protocol within this AS have not learned this routing information, data packets may be discarded, because these routers do not know this route when these packets traverses this AS, which may cause a route black hole.

The BGP-IGP synchronization is designed to ensure all routers within this AS can learn the outgoing routing information. Simply, the BGP Speakers redistribute all of the routes learned by the BGP protocol to the IGP protocol to ensure that the routers within the AS learn such routing information.

The BGP-IGP synchronization mechanism can be cancelled in two situations:

10) There is no routing information passing through the local AS (In general, this AS is an end AS).

11) All routers within this AS run the BGP protocol and a full connection is established among all BGP Speakers (An adjacent relationship is established between any two BGP Speakers).

Caution

By default, synchronization is disabled. Enable synchronization when not all the routers are running BGP when traversing an AS.

To enable synchronization of BGP speakers, execute the following commands in BGP configuration mode:

Command	Function
Ruijie(config-router)# **synchronization**	Enables synchronization of BGP and IGP.

Execute the **no synchronization** command to disable the synchronization mechanism.

Configuring Interaction between BGP and IGP

To inject the routing information generated by the IGP protocol into the BGP protocol, execute the following commands in BGP configuration mode:

Command	Function
Ruijie(config-router)# **redistribute** [**connected** | **rip** | **static**] [**route-map** *map-tag*] [**metric** *metric-value*]	(Optional) Redistributes static route, direct route and the routing information generated by RIP.
Ruijie(config-router)# **redistribute ospf** *process-id* [**route-map** *map-tag*] [metric *metric-value*] [**match internal external** [**1** | **2**] **nssa-external** [**1** |**2**]]	(Optional) Redistributes the routing information generated by OSPF.
Ruijie(config-router)# **redistribute isis** [*isis-tag*] [**route-map** *map-tag*] [**metric** *metric-value*] [**level-1**| **level-1-2**| **level-2**]	(Optional) Redistributes the routing information generated by ISIS.

By default, distribution of a default route is disabled. To enable this function, execute the following command:

Command	Function
Ruijie(config-router)# **default-information originate**	Redistributes the default route.

Configuring BGP Timer

The BGP uses the Kepalive timer to maintain an effective connection with the peers, and takes the Hldtime timer to determine whether the peers are valid. By default, the value of the Kepalive timer is 60s, and the value of the Holdtime timer is 180s. When a BGP session is established between BGP Speakers, both parties will negotiate with the Holdtime timer and the one with a smaller value will be selected. The selection of the Keepalive timer is based on the smaller one between 1/3 of the negotiated Holdtime timer and the configured Keepalive timer.

To adjust the value of the BGP timer based on all peers, execute the following commands in BGP configuration mode:

Command	Function
Ruijie(config-router)# **timers bgp** *keepalive holdtime*	Adjusts the keepalive and holdtime values of BGP based on all peers. The range of the *keepalive* is 0 to 65535 seconds, and 60 seconds by default. The range of the *holdtime* is 0 to 65535 seconds, 180 seconds by default.

Certainly, you can adjust the value of the BGP timer based on the specified peers, and execute the following commands in BGP configuration mode:

Command	Function
Ruijie(config-router)# **neighbor** {*address* | *peer-group-name*} **times** *keepalive holdtime*	Configures the Keepalive and Holdtime value to establish a session with the specified BGP peer (group). The range of the keepalive is 0 to 65535 seconds, 60 seconds by default. The range of the holdtime is 0 to 65535 seconds, 180 seconds by default.

Use the **no** form of the right command to clear the value of configured timer.

Configuring BGP Path Attributes

AS_PATH Attribute

The BGP protocol controls the distribution of routing information in the following ways:

- IP address by using the **neighbor distribute-list** and **neighbor prefix-list** commands
- AS_PATH Attribute (refer to this section)
- COMMUNITY Attribute (refer to the COMMUNITY Attribute configuration)

You can use the AS path-based access control list to control the distribution of the routing information, where the AS path-based ACL will use Regular Expression to resolve the AS path.

To configure the AS path-based distribution of routing information, execute the following commands in privileged EXEC mode:

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **ip as-path access-list** *path-list-name* {**permit** | **deny**} *as-regular-expression*	(Optional) Defines an AS path list.
Ruijie(config)# **ip routing**	Enables the routing function (if disabled)

Command	Function
Ruijie(config)# **router bgp** *as-number*	Enables the BGP and configures this AS number to enter BGP configuration mode.
Ruijie(config-router)# **neighbor** {*address* | *peer-group-name*} **filter-list** *path-list-name* {**in** | **out**}	(Optional) Implements the routing policy according to the AS path list when the routing information is received from and sent to the specified BGP peer (group).
Ruijie(config-router)# **neighbor** {*address* | *peer-group-name*} **route-map** *map-tag* {**in** | **out**}	(Optional) Implements the routing policy according to the route-map when the routing information is received from and sent to the specified BGP peer (group). In route-map configuration mode, you can use the match as-path to operate the AS path attribute based on the AS path list, or take the set as-path to operate the AS attribute value.

The BGP protocol will not consider the length of the AS path when selecting the optimal path as specified in RFC1771. In general, the shorter the AS path, the higher its priority. Hence, we take the length of the AS path as the optimal path. You can determine whether to consider the length of the AS path when selecting the optimal path according to actual condition.

If you wish to ignore the length of the AS path when selecting the optimal path, execute the following commands in BGP configuration mode:

Command	Function
Ruijie(config-router)# **bgp bestpath as-path ignore**	Compares the length of the AS path when selecting the optimal path.

Caution

Within the AS, all BGP Speakers consider the length of the AS path as consistent when selecting the optimal path. Otherwise, the optimal path information selected by different BGP Speakers will be different.

NEXT_HOP Attribute

To set the next hop as the local BGP Speaker for sending the routing information to the specified BGP peer, you can use the **neighbor next-hop-self** command, which is mainly used in non-mesh networks, such as frame relay and X.25. Execute the follwing commands in BGP configuration mode:

Command	Function
Ruijie(config-router)# **neighbor** {*address* | *peer-group-name*} **next-hop-self**	Sets the next hop as the local BGP speaker for distributing the routing information to the specified BGP peer (group).

You can also modify the next hop of the specified path by using the **set next-hop** command of Route-map.

Caution

This command is not recommended for a fully meshed network such as Ethernet, because it may cause additional hops and incur unnecessary overhead.

MULTI_EXIT_DISC Attribute Configuration

The BGP takes the MED value as the foundation for priority comparision of the paths learned from the EBGP Peers. The smaller the MED value, the higher the path priority.

By default, the protocol only compares it with the MED value for the path of the peers from the same AS when the optimal path is selected. If you hope to compare it with the MED value for the path of the peers from different ASs, execute the following commands in BGP configuration mode:

Command	Function
Ruijie(config-router)# **bgp always-compare-med**	Compares with the MED value for the path of different ASs.

By default, it will not compare with the MED value for the path of the peers for other ASs within the AS when the optimal path is selected. If you hope to compare with the MED value for the path of the peers from different AS confederations, execute the following commands in the BGP configuration mode.

Command	Function
Ruijie(config-router)# **bgp bestpath med confed**	Compares with the MED value for the path of the peers from other ASs within the confederation.

By default, if a path with an undefined MED attribute is received, the MED value of this path will be taken as 0. The smaller the MED value, the higher the path priority. The MED value of this path reaches the highest priority. If you want the MED attribute for the path with undefined MED attribute to present the lowest priority, execute the following command in BGP configuration mode:

Command	Function
Ruijie(config-router)# **bgp bestpath med missing-as-worst**	Sets the priority of the path whose MED attribute is not set as the lowest.

By default, they are compared with each other in the sequence the paths are received when the optimal path is selected. If you want to first compare with the path of the peers from the same AS, execute the following commands in BGP configuration mode:

Command	Function
Ruijie(config-router)# **bgp deterministic-med**	Compares first with the path of the peers from the same AS. By default, they will be compared with by the receiving sequence. The later received path will be compared with first.

LOCAL_PREF Attribute Configuration

The BGP takes the LOCAL_PREF as the foundation for priority comparision of the path learned from the IBGP peers. The larger the LOCAL_PREF value, the higher the path priority.

The BGP Speakers will add the local preference when they send the received external routes to the IBGP peers. To modify the local preference, execute the following commands in BGP configuration mode:

Command	Function
Ruijie(config-router)# **bgp default local-preference** *value*	Changes the default local preference. The range of the value is 0 to 4294967295, 100 by default.

You can also modify the local preference of the specified path by using the **set local-preference** command of Route-map.

COMMUNITY Attribute Configuration

COMMUNITY Attribute provides another way to control the distribution of the routing information.

The community is a set of destinations. The purpose is to implement the community-based routing policy so as to simplify the configuration for the distribution of the routing information in the BGP Speakers.

Each destination may have more than one community, and the manager of the AS can define the community destination.

By default, all destinations belong to the Internet community carried in the community attribute of the path.

At present, totally four common community attributes are predefined:

- **Internet**: Indicates the Internet community, and all paths are in this community.
- **no-export**: Indicates this path will not be exported to BGP peers.
- **no-advertise**: Indicates this path will not be advertised to BGP peers.
- **local-as**: Indicates this path will be advertised only in the local AS or the AS confederation if it is configured.

You can control the receiving, priority and distribution of the routing information by using the community attribute.

The BGP supports up to 32 COMMUNITY attributes for every route. When configuring the **route-map** command, you can set up to 32 COMMUNITY attributes for the parameters **match** and **set COMMUNITY.**

The BGP Speakers can set, add or modify the community attribute value when they learn, issue or redistribute a route. The aggregated path includes the community attribute of all aggregated paths when route aggregation is carried out.

To configure the community attribute-based distribution of routing information, execute the following commands in privileged EXEC mode:

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.

Command	Function
Ruijie(config)# **ip community-list standard** *community-list-name* {p**ermit** | **deny**} *community-number*	(Optional) Creates the community list. The *community-list-name* is the name of the community list. The community-number is the concrete value of the community list in the range 1 to 4,294,967,295, or the well-known community attribute such as Internet, local-AS, no-advertise and no-export.
Ruijie(config)# **ip routing**	Enables the routing function (if disabled).
Ruijie(config)# **router bgp** *as-number*	Enables the BGP and configure this AS number to enter into BGP configuration mode.
Ruijie(config-router)# **neighbor** {*address* | *peer-group-name*} **send-community**	(Optional) Configures the router to send the community attribute to the specified BGP peer (group).
Ruijie(config-router)# **neighbor** {*address* | *peer-group-name*} **route-map** *map-tag* {**in** | **out**}	(Optional) Configures the router to implement the routing policy according to the route-map when the routing information is received from and sent to the specified BGP peer (group). In the route-map configuration mode, you can use the match community-list [exact] and set community-list delete to operate the community attribute by the community list, or use the set community command to operate the community attribute value directly.

Other Related Configuration

By default, if two paths with identical path attributes are received from different EBGP peers during the selection of the optimal path, we will select the optimal path based on the path receiving sequence. You can select the path with a smaller router ID as the optimal path by using the following commands.

Command	Function
Ruijie(config-router)# **bgp bestpath compare-routerid**	Allows the BGP to compare with the router ID when the optimal path is selected.

Selecting the Optimal Path for BGP

Optimal route selection forms an important part of the BGP protocol. The following section describes the selection process of the BGP route protocol in detail:

- An invalid routing table entry is not allowed in the selection of optimal routes.

Caution Invalid entries include those unreachable for the next hop and those in oscillation.

- Select the route with the high LOCAL_PREF attribute value.
- Select the route generated by the local BGP speaker.

The route generated by the local BGP speaker includes the one generated by the **network, redistribute, aggregate** command.

- Select the route with the shortest AS length.
- Select the route with the lowest ORIGIN attribute.
- Select the route with the smallest MED value.
- The EBGP path has a higher priority than the IBGP path and the AS confederation, and the priority is identical for the IBGP path and the AS confederation.
- Select the route with the smallest IGP metric to reach the next hop.
- Select the route received earlier from the EBGP routes.
- Select the route which advertises that the router ID of the BGP speaker is small.
- Select the route with the greater cluster length.
- Select the route: the value of neighbor address for which is high.

Caution

Discussed above is the process of select the optimum route under the default configuration. You can change the selection process of the route by the CLI command. For instance, you can use the **bgp bestpath as-path ignore** command to make step 4 part of the process of invalidating the optimal route. Use the bgp bestpath compare-routerid command to invalidate Step 9 of the selection.

Configuring BGP Route Aggregation

Since the BGP-4 supports CIDR, aggregated entries can be created to downsize the BGP routing table. Certainly, BGP aggregated entries can be added to the BGP routing table only when there is a valid path within the aggregation scope.

To configure the BGP route aggregation, execute the following commands in BGP configuration mode:

Command	Function
Ruijie(config-router)# **aggregate-address** *address mask*	(Optional) Configures the aggregated address.
Ruijie(config-router)# **aggregate-address** *address mask* **as-set**	(Optional) Configures the aggregated address, and remain the AS path information of the path within the scope of the aggregated address.
Ruijie(config-router)# **aggregate-address** *address mask* **summary-only**	(Optional) Configures the aggregated address and only advertise the aggregated path.
Ruijie(config-router)# **aggregate-address** *address mask* **as-set summary-only**	(Optional) Configures the aggregated address, and remain the AS path information of the path within the scope of the aggregated address. At the same time, only the aggregated path is advertised.

Use the **no** form of the above commands to disable the configuration.

Caution

By default, the BGP will advertise all routing information both before and after aggregation. If you want to advertise only the aggregated path information, use the **aggregate-address summary-only** command.

Configuring Route Reflector for BGP

To speed up the convergence of routing information, all BGP Speakers within one AS will usually establish the full connection (The adjacent relationship is established between any two BGP Speakers). Too many BGP Speakers within the AS may increase the resource overhead of the BGP Speakers, raise the configuration workload and complexity of network administrators, and reduce the network scalability.

Therefore, route reflector and AS confederation are preferrably used to reduce the connections of the IBGP peers within an AS.

The route reflector provides a way to reduce the connections of the IBGP peer within the AS. One BGP Speaker is set as the route reflector, which divides the IBGP peer within this AS into two types, such as client and non-client.

The rule to implement the route reflector within the AS is shown as follows:

- Configure the route reflector and specify its client, so the route reflector and other clients form a cluster. The route reflector establishes the connection with clients.
- The clients of the route reflector within one cluster should not establish the connection with other BGP Speakers of other clusters.
- Within an AS, a full connection is established among the IBGP peer of non-clients. The IBGP peer of non-clients involves the following scenarios: among several route reflectors within one cluster, among the route reflector within the cluster and the BGP Speakers not involved in the route reflector function out of the cluster (In general, the BGP Speakers don’t support the route reflector function), among the route reflector within the cluster and the route reflector of other cluster.

The following rules applies when the route reflector receives one route:

- The route update received from the EBGP Speaker will be sent to all clients and non-clients.
- The route update received from the clients will be sent to other clients and all non-clients.
- The route update received from the IBGP non-clients will be sent to all its clients.

To configure the BGP route reflector, execute the following commands in BGP configuration mode:

Command	Function
Ruijie(config-router)# **neighbor** {*address* | *peer-group-name*} **route-reflector-client**	Configures this product as the route reflector and specifies its clients.

In general, one group is only configured with one reflector. In this case, the Router ID of the route reflector can be used to identify this cluster. To increase the redundancy, you can set more than one route reflector within this cluster. You must configure the cluster ID, so that one route reflector can identify the route update from other route reflectors of this cluster.

Caution To set several route reflectors for one cluster, you need to configure a cluster ID for this cluster.

To configure the cluster ID of the BGP, execute the following commands in BGP configuration mode:

Command	Function
Ruijie(config-router)# **bgp cluster-id** *cluster-id*	Configures the cluster ID of the route reflector.

In general, it is not necessary to establish a connection between the clients of the route reflector within the cluster, as the route reflector will reflect the routes among clients. However, this function can be disabled if a full connection is established among all clients.

To disable the function of reflecting the client routes, execute the following commands in BGP configuration mode:

Command	Function
Ruijie(config-router)# **no bgp client-to-client reflection**	Disables route reflection on clients.

Configuring Route Flap Dampening for BGP

Route flap means that a route changes between the valid status and the invalid status. The route flap usually causes instable routes to be transmitted on the Internet, thus resulting an unstable network. BGP route flap dampening provides a way to reduce route flap by monitoring the routing information of EBGP peers.

The route flap dampening of BGP uses the following terminologies:

- Route Flap: A route changes between the valid status and the invalid status.
- Penalty: The route flap dampening-enabled BGP Speakers will add a penalty for the route each time when a route flaps. The penalty will be accumulated to exceed the suppress limit.
- Suppress Limit: When the penalty of a route exceeds this value, the route will be suppressed.
- Half-life-time: The time elapsed when the penalty is reduced to half of its value.
- Reuse Limit: When the penalty of the route is lower than this value, route suppression is released.
- Max-suppress-time: The maximum amount of time the route can be suppressed.

Overview of route flap dampening: The BGP Speakers will add a penalty for the route each time when a route flaps. The penalty is accumulated. Once the penalty value reaches the suppress limit, the route will be suppressed. When the half-life-time is reached, the penalty value is reduced to half of its value. Once the penalty value is reduced to the reuse limit, the route will be activated again. A route can be suppressed for the maximum suppress time.

To configure the route flap dampening of the BGP, execute the following commands in BGP configuration mode:

Command	Function
Ruijie(config-router)# **bgp dampening**	Enables the route flap dampening of the BGP protocol.
Ruijie(config-router)# **bgp dampening** *half-life-time reuse suppress max-suppress-time*	(Optional) Configures the parameters of the route flap dampening. half-life-time: in the range of 1 to 45minutes, 15minutes by default. reuse: in the range of 1 to 20000, 750 by default. suppress: in the range of 1 to 20000, 2000 by default. max-suppress-time: in the range of 1 to 255 minutes, 4*half-life-time by default.

To monitor the route flap dampening information if necessary, execute the following commands in privileged EXEC mode:

Command	Function
Ruijie# **show ip bgp dampening flap-statistics**	(Optional) Shows the flap statistics information of all routers.
Ruijie# **show ip bgp dampening dampened-paths**	(Optional) Shows the dampened statistics.

To clear the route flap dampening information or the dampened routes, execute the following commands in BGP configuration mode:

Command	Function
Ruijie# **clear ip bgp flap-statistics**	(Optional) Clears flap statistics about all un-dampened route.
Ruijie# **clear ip bgp flap-statistics** address mask	(Optional) Clears flap statistics about the specified route (excluding the dampened routes).
Ruijie# **clear ip bgp dampening** [*address mask*]	(Optional) Clears flap statistics about all routes, and releases the suppressed routes.

Configuring AS Confederation for BGP

Confederation provides a way to reduce the connections of the IBGP peer within the AS.

One AS is divided into multiple sub ASs that can form a confederation by setting a unified confederation ID (namely, confederation AS number). An external confederation is still considered an AS and only the AS number of the confederation is visible. Within the confederation, the full IBGP peer connection is still established among the BGP Speakers, and the EBGP connection is established among the BGP Speakers within the sub AS. Although the EBGP connection is established among BGP Speakers within the sub ASs, the path attribute information of NEXT_HOP, MED and LOCAL_PREF remains intact when the information is exchanged.

To implement the AS confederation, execute the following commands in BGP configuration mode:

Command	Function
Ruijie(config-router)# **bgp confederation identifier** *as-number*	Configures the AS confederation number. The range of *as-number* is 1 to 4294967295.
Ruijie(config-router)# **bgp confederation peers** *as-numbe* [as-number..]	Configures other sub AS numbers within the AS confederation. The range of *as-number* is 1 to 4294967295.

Use the **no** form of the above commands to disable the configuration.

Configuring BGP Management Distance

The management distance indicates the reliability of the routing information resource, within the range of 1 to 255. The larger the value of the management distance, the lower the reliability is.

The BGP sets different management distances for different information sources that have been learned, such as External-distance, Internal-distance and Local-distance.

- **External-distance:** The management distance of the route learned from the EBGP peers.
- **Internal-distance:** The management distance of the route learned from the IBGP peers.
- **Local-distance:** The management distance of the route learned from the peers. However, the optimal one can be learned from the IGP. In general, these routes are indicated by the **Network Backdoor** command.

To modify the management distance of the BGP protocol, execute the following commands in BGP configuration mode:

Command	Function
Ruijie(config-router)# **distance bgp** *external-distance internal-distance local-distance*	Configures the management distance. The range of the distance is 1 to 255. For the default configuration: *external-distance* *20* *internal-distance* *200* *local-distance* *200*

Use the **no** form to restore the default management distance of the BGP protocol.

Caution

It is not recommended that you change the management distance of the BGP route. If the change is necessary, please make sure:

■ The External-distance is lower than the management distance of other IGP route protocol (OSPF and RIP).

■ The Internal-distance and Local-distance is higher than the management distance of other IGP route protocol.

Configuring BGP Route Update Mechanism

The BGP route update mechanism comprises two parts: timing scanning update and event trigger update. The former means that the timer is used in the BGP to start the scanning mechanism periodically to update the routing table. The latter means that when BGP configuration or the next hop of BGP route changes, the BGP protocol starts the scanning mechanism to update the routing table.

To configure the BGP route update mechanism, execute the following commands in BGP configuration mode:

Command	Function
Ruijie(config-router)# **bgp scan-rib disable**	Enables the event trigger mechanism. By default, the timing scanning update mechanism is used.
Ruijie(config-router)# **bgp scan-time** *scan-time*	(Optional) Sets the scanning interval. *scan-time*: In the range of 5 to 60 seconds, 60 seconds by default

You can also configure this command in IPv4/IPv6/VPNv4/IPv4 vrf address family mode.

Use the **no** form to remove the configuration.

Caution

When you run the **bgp scan-rib disable** command to enable the event trigger mechanism, the synchronization should be disabled and the BGP next hop trigger mechanism should be enabled. When synchronization is enabled or the BGP next hop trigger mechanism is disabled, the BGP updates the routing table in timing scanning mode.

Configuring BGP Nexthop Trigger Update Mechanism

The BGP next hop trigger update mechansim improves the converge of BGP routes. It monitors the next hop of BGP routes to speed up converge in stable network topology.

By default, the BGP next hop trigger update mechansim is enabled. After establishing connections with neighbors, the BGP will automatically monitor the next hops of the routes learned from neighbors. When the next hop changes, the BGP will receive a notification of updating the routing tble. This can reduce the time to check the change of next hop for better converge of BGP routes.

If the function is disabled, scan-timer will periodically scan updates to the next hope of BGP.

To configure the BGP next hop trigger update mechanism, execute the following commands in BGP configuration mode:

Command	Function
Ruijie(config-router)# **bgp nexthop trigger enable**	Enables the function of triggering the next BGP route. This function is enabled by default.
Ruijie(config-router)# **bgp nexthop trigger delay** ***delay***-*time*	(Optional) Sets the delay of the BGP next hop trigger update. *delay-time*: In the range of 0 to 100 seconds, 5 seconds by default

You can also configure this command in IPv4/IPv6/VPNv4/IPv4 vrf address family mode.

Use the **bgp nexthop trigger enable** command to restore the setting to the default value.

The **bgp nexthop trigger enable** command and the bgp scan-time command control the same timer. When the timing scanning mechanism is enabled (bgp scan is enabled by default. The **bgp scan-rib disable** command is used to disable **bgp scan**）， the time of larger than 60 seconds set by the **bgp nexthop trigger enable** command does not take effect because the timing scanning mechanism is always activated before the delay time.

Caution

In an unstable network (the next hop changes frequently), especially when there are a large number of routes, this function carries out unnecessary route calculation and consumes more CPU resources. In this case, it is recommended that you disable the BGP next hop trigger update mechanism.

Configuring BGP GR

GR (Graceful Restart) can ensure continuous data forwarding during the resetting of the BGP protocol. Currently, Ruijie supports GR during active and standby switching on its high-end devices to ensure service continuity.

Working Mechanism of GR

12) Standard

RFC4724: Graceful Restart Mechanism for BGP, which is represented by BGP GR later.

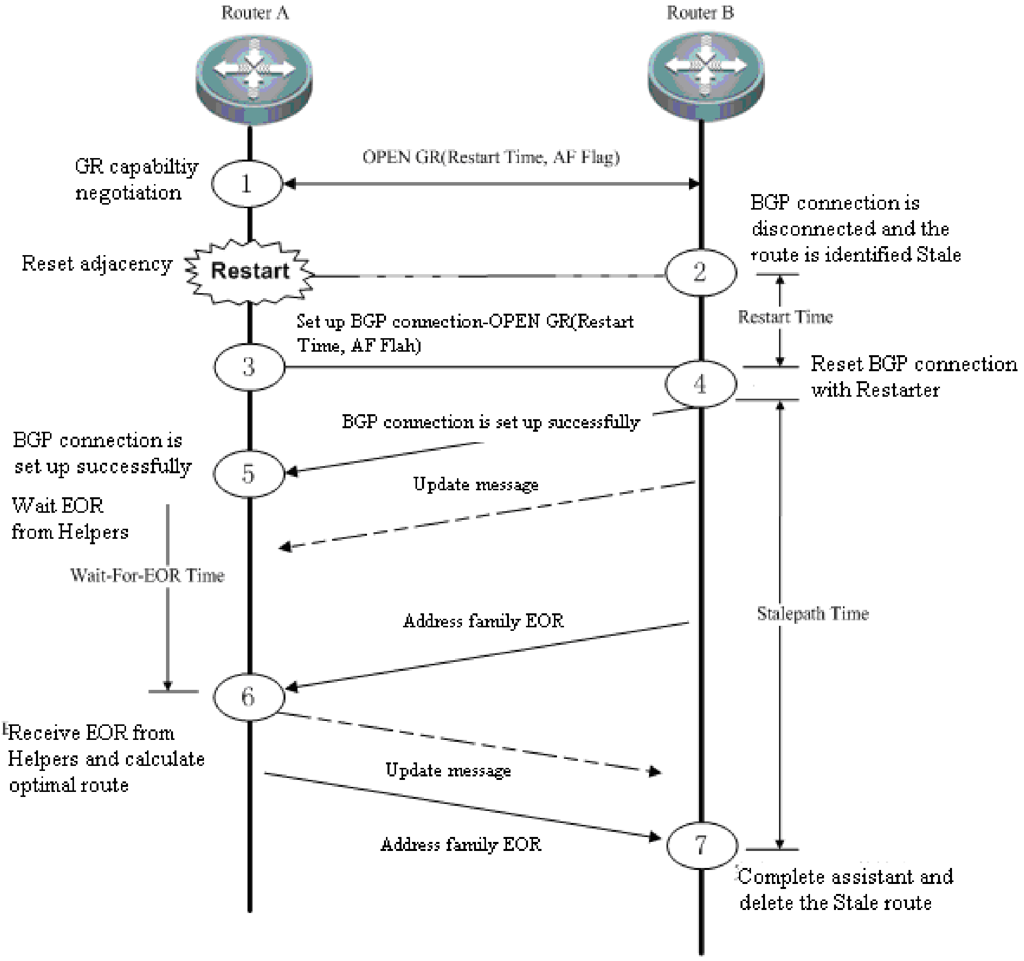
13) Working mechanism

RFC4724 is a standard GR protocol that IETF especially defines for the BGP protocol. This document outlines the principles of BGP GR, including:

- Graceful Restart Capability is added to the OPEN message of the BGP protocol, indicating that the BGP supports GR. The GR capability is negotiated by neighbors during the initiation of BGP connection.
- GR Restarter and GR Helper. GR Restarter means that the router restarts the BGP protocol, which can ensure continuous route forwarding when the route control panel fails. GR Helper is the BGP neighbor of the GR Restarter that assists the GR Restarter in BGP GR for continous forwarding across the network.
- In the update message, EOR (End-of-RIB) is added to indicate that the route message update is complete.

The following ffigure illustrates the process of BGR GR.

Figure 1 Process of graceful restart for BGP


Router A
Router B
GR capabiltiy negotiation
1
OPEN GR(Restart Time, AF Flag)
BGP connection is disconnected and the route is identified Stale
Reset adjacency
Restart
2
Restart Time
Set up BGP connection-OPEN GR(Restart Time, AF Flah)
3
4
Reset BGP connection with Restarter
BGP connection is set up successfully
BGP connection is set up successfully
5
Update message
Wait EOR from Helpers
Wait-For-EOR Time
Stalepath Time
Address family EOR
6
Receive EOR from Helpers and calculate optimal route
Update message
Address family EOR
7
Complete assistant and delete the Stale route

Initially, the BGP protocol establishes a adjacency and negotiates respective GR capability with the GR Capability field of the OPEN message. At a point, the device reboots and the BGP session is disconnected. The neighbor detects disconnection. With GR supported, the BGP neighbor keeps the route of the GR Restarter valid but identifies it in Stale (aged, not updated) state. The GR Restarter reboots and re-establishes connection with the GR Helper and waits the route update message and EOR label from the GR Helper. After receiving an EOR label from all neighbors, the BGP Restarter calculates routes and update the routing table, and begins to send update routes to the GR Restarter. Upon the receipt of these routes, the GR Helper removes the Stale tag from these routes and then deletes the routes (these routes have not been updated) tagged with Stale after receiving the EOR label from the BGP Restarter, calculates routes and updates the routing table.

Some key timers are defined to assist the implemetation of BGP GR:

- **Restart-Timer**: The GR Restarter notifies the GR Helper of restart time that the GR Helper needs to wait before reestablishing the BGP connection. You can modify this value by using the **bgp graceful-restart restart-time** command.
- **Wait-For-EOR Timer**: Time the GR Restarter needs to wait for the EOR label of all GR Helpers. After receiving the EOR label of all GR Helpers or the timer expires, the GR Restarter calculates optimal routes and updates the routing table. You can modify this value by using the **bgp update-delay** command.
- **StalePath Timer**: Time the GR Helper needs to wait before receiving the EOR label from the GR Restarter after reestablishing the connection with the GR Restarter. During this period, the GT Helper keeps alive the route of the

GR Restarter. It will delete the rotue tagged with Stale after receiving the EOR lable or the StalePath timer expires. You can modify this value by using the **bgp graceful-restart stalepath-time** command.

Implementation of BGP GR

Implementation of BGP GR is not an independent process. All BGP peers are necessary to enable BGP GR capability for normal operation. Failed GR may cause a temporary route black hole or loop and affect the network operation. Consequently, it is recommended that you ensure the GR capability is negotiatied successfully by using the **show ip bgp neighbors** command. To enable BGP GR, exeucte the **bgp graceful-restart** command in BGP route configuration mode.

Configuring BGP GR Capability

BGP GR capability is an extended capability of the BGP protocol, which is disabled by default. When enabling GR, the BGP reestablishes the connection with its neighbor and negotiates GR capability. The GR is enabled only when both sides support GR capability.

To enable the GR capability, execute the following commands:

Command	Function
Ruijie # **configure terminal**	Enters global configuration mode.
Ruijie(config)# **router bgp** *500*	Enters BGP configuration mode.
Ruijie(config-router)# **bgp graceful-restart**	Enables GR.
Ruijie(config-router)# **end**	Returns to privileged EXEC mode.
Ruijie # **show running-config**	Shows the configuration.
Ruijie # **write**	(Optional) Saves the configuration.

☑ All the BGP-enabled products support this command.

Caution

The **bgp graceful-restart** command does not take effect for established BGP connections. Namely, the BGP connection will not negotiate GR capability immediately when it is in Established status. In this case, you need to forcibly restart the peer to renegotiate the GR capability, for instance, **clear ip bgp 192.168.195.64**. This is to prevent the restart of neighboring relations for capability renegotiation when GR is enabled or disabled, as renegotiation may cause network oscillation. Therefore, you can decide whether to restart neighbors.

Supporting BGP GR capability does not mean a device can serve as the GR Restarter for graceful restart, which also depends on the hardware. The GR Restarter device of Ruijie Networks needs to support dual-engine redundant hot backup.

Configuring BGP GR Timer

After the GR capability is enabled, the BGP automatically configures relevant timers with default values. By default, the Restart Timer is 120s, the Wait-For-EOR Timer is 120s and the StalePath Timer is 360s.

To configure these timers, execute the following commands:

Command	Function
Ruijie # **configure terminal**	Enters global configuration mode.
Ruijie(config)# **router bgp** *500*	Enters BGP configuration mode.
Ruijie(config-router)# **bgp graceful-restart**	Enables GR capability.
Ruijie(config-router)# **bgp graceful-restart restart-time** *150*	Sets the Restart Timer to 150s.
Ruijie(config-router)# **bgp update-delay** *150*	Sets the Wait-For-EOR Timer to 150s.
Ruijie(config-router)# **bgp graceful-restart stalepath-time** *400*	Sets the StalePath Timer to 400s.
Ruijie(config-router)# **end**	Returns to privileged EXEC mode.
Ruijie # **show running-config**	Shows the configuration.
Ruijie # **write**	(Optional) Saves the configuration.

☑ All the BGP-enabled products support this command.

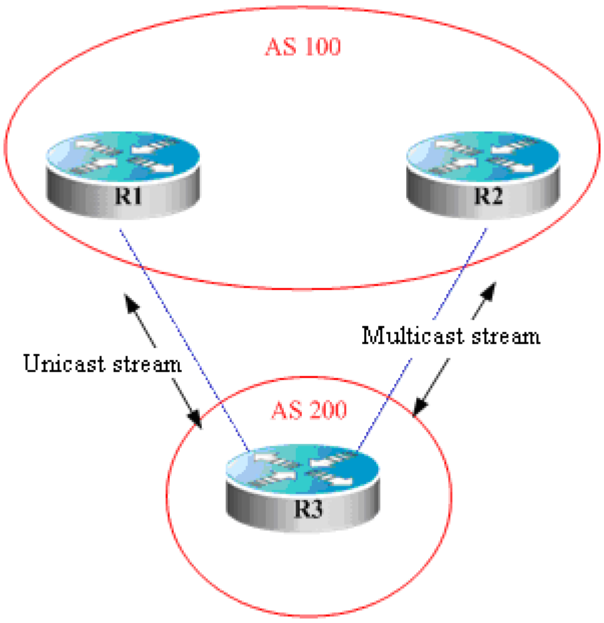
⚠ Caution

The restart time configured by using the **bgp graceful-restart restart-time** command should not exceed the Hold time of the BGP peer. Otherwise, the Hold Time will be used as the restart time and be notified to the peer for GR capability negotiation.

Configuring BGP Multicast

The BGP multicast route is used for multicast RFC check. In general, the multicast forwarding topology is similar to the unicast forwarding topology. You can design different multicast topologies by using BGP multicast, which is used for the multicast topology between ASs, as shown in the following figure.

Figure 2



There are two routers in AS100. In terms of design, unicast streams are sent to R1 and multicast streams to R2. In this case, MPBGP is required between R2 and R3.

Step 1: Enable BGP on R1, R2 and R3 and establish neighbors among them.

Take R3 as an example. Configures R1 and R2 as its BGP neighbors.

Command	Function
Ruijie # **configure terminal**	Enters global configuration mode.
Ruijie(config)# **router bgp** *200*	Enters BGP configuration mode with the AS number of 200.
Ruijie(config-router)# **neighbor** *R2* **remote-as** *100*	Configures R2 as the BGP neighbor with the AS number of 100.
Ruijie(config-router)# **neighbor** *R1* **remote-as** *100*	Configures R1 as the BGP neighbor with the AS number of 100.

Step 2: Since R3 does not need to transmit multicast routes with R1, disable the multicast address of R1.

Command	Function
Ruijie(config-router)# **address-family ipv4 multicast**	Enters the IPv4 multicast address family configuration mode.
Ruijie(config-router-af)# **no neighbor** *R1* **active**	Disable the multicast address of R1.

Step 3: Since R2 needs to transmit multicast rotues with R1, enable the multicast address of R2.

Command	Function
Ruijie(config-router)# **address-family ipv4 multicast**	Enters IPv4 multicast address family configuration mode.
Ruijie(config-router-af)# **neighbor** *R2* **active**	Enables the multicast address of R2.

Step 4: Import the routes that R3 needs to advertise to R2 in multicast address family mode.

Routes are imported by using **redistribute**, network advertising, and aggregate route publishing. They are configured in a multi-address family, for example:

Command	Function
Ruijie(config-router)# **address-family ipv4 multicast**	Enters IPv4 multicast address family configuration mode.
Ruijie(config-router-af)# **redistribute ospf** 1	Redistributes OSPF routes.

Caution

During the process of redistribution, the routes imported are unicast routes. For instance, the **redistribute ospf 1** command imports OSPF unicast routes. This is because that multicast routes depend on the egress of unicast routes for the establishment of a multicast spanning tree.

Configuring BGP Local AS

This function configures a local AS different from the real AS (rotuer BGP AS) for one peer, which is equivalent to virtualizing an AS. When the real AS changes, you still can establish a BGP connection without modifying the BGP configuration of the peer. Local AS applies to AS migration and converge of large networks without affecting the configurations of the devices in other interconencted ASs.

When establishing the BGP connection, the local device will advertise the local AS number to the peer in an OPEN message. The peer checks whether the AS number matches the local one and rejects the BGP connection if there is a difference. By default, the local AS of the BGP connection is the real BGP AS. With this function, the local device replaces the real AS with the configured one to establish a BGP connection.

By default, no peer is configured with Local AS. The Local AS of the peer is real AS over BGP. To configure a local AS for one peer, execute the following commands:

Command	Function
Ruijie # **configure terminal**	Enters global configuration mode.
Ruijie(config)# **router bgp** *500*	Enters BGP configuration mode.
Ruijie(config-router)# **neighbor** *192.168.195.64* **remote-as** *100*	Configures the peer.
Ruijie(config-router)# **neighbor** *192.168.195.64* **local-as** *300*	Configures AS 300 as the local AS for the peer

The local AS function applies only to EBGP peers, instead of IBGP peers, confederation EBGP peers. Meanwhile, there are some limitations as described below:

- The local AS cannot be configured as the remote peer.
- Local AS cannot be configured for one member of the peer group.
- The local AS cannot be configured as the real BGP AS.
- The local AS cannot be configured as the AS number of the confederation if the device is a member of the confederation.

For details about the **neighbor** *peer-address* **local-as** *as-num* command, refer to the *Command Reference*.

Monitoring BGP

You can use the **Shows** commands to view the BGP route table, buffer and database. Execute the follwing commands in privileged EXEC mode:

Command	**Function**
Ruijie# **show ip bgp**	Shows the information on all BGP routes.
Ruijie# **show ip bgp** {*network* | *network-mask* } [longer-prefixes]	Shows the BGP routing information of the specified destination.
Ruijie# **show ip bgp prefix-list** *prefix-list-name*	Shows the BGP routing information of the specified matching against the prefix list.
Ruijie# show **ip bgp community** [**exact**] *community-number*	Shows the BGP routing information including the specified community.
Ruijie# **show ip bgp community-list** *community-lister-number* [exact]	Shows the BGP routing information which matches against the specified community list.
Ruijie# **show ip bgp filter-list *path-list***-*number*	Shows the BGP routing information which matches against the specified AS path list.
Ruijie# **show ip bgp regexp** *as-regular-expression*	Shows the BGP routing information of the specified regular expression which matches against the AS path attribute.
Ruijie# **show ip bgp dampening dampened-paths**	Shows the suppressed flap statistics information.
Ruijie# **show ip bgp dampening flap-statistics**	Shows the flap statistics information of all routes with the flap record.
Ruijie# **show ip bgp neighbors** [*address*] [**received-routes** | **routes** | **advertised-routes** | **received]**	Shows the information of the BGP peer.
Ruijie# **show ip bgp summary**	Shows the configuration of the BGP router and the information about the peer.
Ruijie# show **ip bgp peer-group** [*peer-group-name*]	Shows the configuration of the BGP peer group.

Protocol Independent Configuration

route-map Configuration

The BGP protocol follows the Route-map policy. For detailed configurations, refer to the Protocol Independent Configuration.

Regular Expression Configuration

The regular expression is a formula used to match the string based on a template. The regular expression is used to evaluate the text data and return a true or false value, that is to say, it determines whether the expression can describe this data correctly.

Description of Control Characters for Regular Expression

The BGP path attribute uses the regular expression. The following table describes the use of the special characters for the regular expression:

Characters	Signs	Special Functions
Period	.	Matched with any single character.
Asterisk	*	Matched with none or any sequence of the string.
Plus	+	Matched with one or any sequence of the string.
Interrogation Mark	?	Matched with none or one sign of the string.
Plus Sign	^	Matched with the starting of the string.
Dollar	$	Matched with the end of the string.
Underlining	_	Matched with the comma, bracket, the starting and end of the string and blank.
Square Brackets	[ ]	Matched with the single character within the specified scope.

Application Example of Regular Expression

Run the **show ip bgp** command on the device:

```
Ruijie# show ip bgp
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
Status Network           Next Hop     Metric  LocPrf  Path
------ ------------------  ---------------   ---------  ---------  -------------------
*>  211.21.21.0/24   110.110.110.10  0     1000   200 300
*>  211.21.23.0/24   110.110.110.10  0     1000   200 300
*>  211.21.25.0/24   110.110.110.10  0     1000   300
*>  211.21.26.0/24   110.110.110.10  0     1000   300
*> 1.1.1.0/24      192.168.88.250  444        0   606
*> 179.98.0.0      192.168.88.250  444        0   606
*> 192.92.86.0     192.168.88.250  8883        0   606
*> 192.168.88.0    192.168.88.250  444         0   606
*> 200.200.200.0   192.168.88.250  777         0   606
```

Use the regular expression in the **show** command:

```
Ruijie# show ip bgp regexp _300_
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
Status Network        Next Hop          Metric  LocPrf  Path
------ ------------------  ---------------   ---------  ---------  -------------------
*>  211.21.21.0/24  110.110.110.10  0     1000   200 300
*>  211.21.23.0/24  110.110.110.10  0     1000   200 300
*>  211.21.25.0/24  110.110.110.10  0     1000   300
*>  211.21.26.0/24  110.110.110.10  0     1000   300
```

BGP Load Protection Configuration

Too many BGP routes may overload a switch, especially a switch with a small memory size. BGP load protection can prevent unforeseen switch problems caused by switch resource usage.

Limiting BGP Routes

To limit BGP routes, configure the maximum number of routes in the BGP address-family mode. Configure the maximum number of routes learned from a BGP neighbor.

Use the following commands to configure the maximum number of routes learned from a BGP neighbor:

Command	Function
Ruijie(config)# **router bgp** *as-num*	Enters BGP configuration mode.
Ruijie(config-router)#**neighbor** {*address* | *peer-group-name*} **remote**-as *as-num*	Configures the BGP neighbor.
Ruijie(config-router)# **neighbor** {*address* | *peer-group-name*} **maximum-prefix** *maximum* [**threshold**] [**warning-only**]	Configures the maximum number of routes learned from the BGP neighbor.

Use the following commands to configure the maximum number of routes in the specified BGP address-family mode:

Command	Function
Ruijie(config)# **router bgp** *as-num*	Enters BGP configuration mode.
Ruijie(config-router)#**address-family ipv4 unicast**	Enters the BGP ipv4 unicast address-family mode.
Or： Ruijie(config-router)#**address-family ipv4** vrf *vrf-name*	Enters the BGP ipv4 VRF address-family mode.
Or： Ruijie(config-router)#a**ddress-family vpnv4 unicast**	Enters the BGP VPNV4 address-family mode.
Ruijie(config-router)# **maximum-prefix** *maximum*	Configures the maximum number of routes in the specified BGP address-family mode.

Configuring Overflow Memory-lack

BGP can be in the overflow state when the memory is insufficient. In OVERFLOW mode, BGP generates a default route to the NULL interface. A newly learned route will be discarded if it is not a default route in the current routing table. In general, the routes BGP learned in the overflow state are dropped, and the system memory stays in a steady state to protect the network from routing loops. In other words, BGP is safe and also preferred in OVERFLOW state.

Use the following commands to move BGP into the overflow state:

Command	Function
Ruijie(config)# **router bgp** *as-num*	Enters BGP configuration mode.
Ruijie(config-router)# **overflow memory-lack**	Brings BGP into the overflow state when memory is running short.

Note

By default, BGP switches to OVERFLOW state automatically when the memory is running short. Use the **no overflow memory-lack** command for the BGP to exit the OVERFLOW state.

Caution

In OVERFLOW state, BGP supports the **clear bgp** { *addressfamily* | **all** } * command. Alternatively, you can disable and re-enable BGP to exit the OVERFLOS state. When the memory becomes sufficient, BGP exits the OVERFLOW state automatically.

BGP Configuration Examples

The following section lists BGP configurations.

Configuring BGP Neighbor

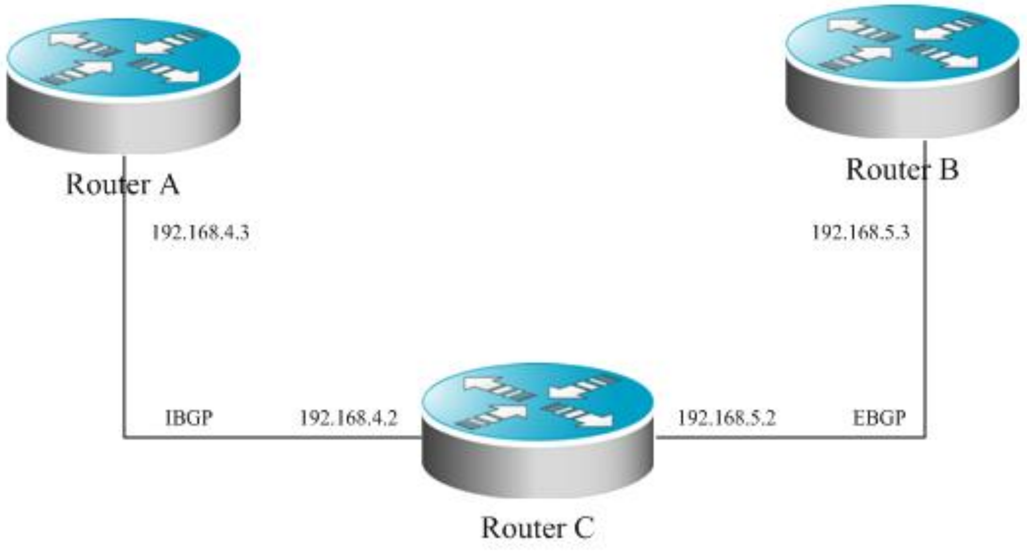
The following section shows how to configure a BGP neighbor. Use the **neighbor remote-as** command to configure the BGP neighbor. Configuration details are shown as follows:

```
router bgp 109
neighbor 131.108.200.1 remote-as 167
neighbor 131.108.234.2 remote-as 109
neighbor 150.136.64.19 remote-as 99
```

Configure one IBGP peer 131.108.234.2 and two EBGP peers 131.108.200.1 and 150.136.64.19.

The following example shows how to configure the BGP neighbor. For the relationship among routers and the assignment of IP addresses, see the figure.

Figure 3



This example shows the BGP configuration of different routers:

Router A configuration:

```
!
router bgp 100
 neighbor 192.168.4.2 remote-as 100
```

Router B configuration:

```
!
router bgp 100
 neighbor 192.168.4.3 remote-as 100
 neighbor 192.168.5.3 remote-as 200
```

Router C configuration:

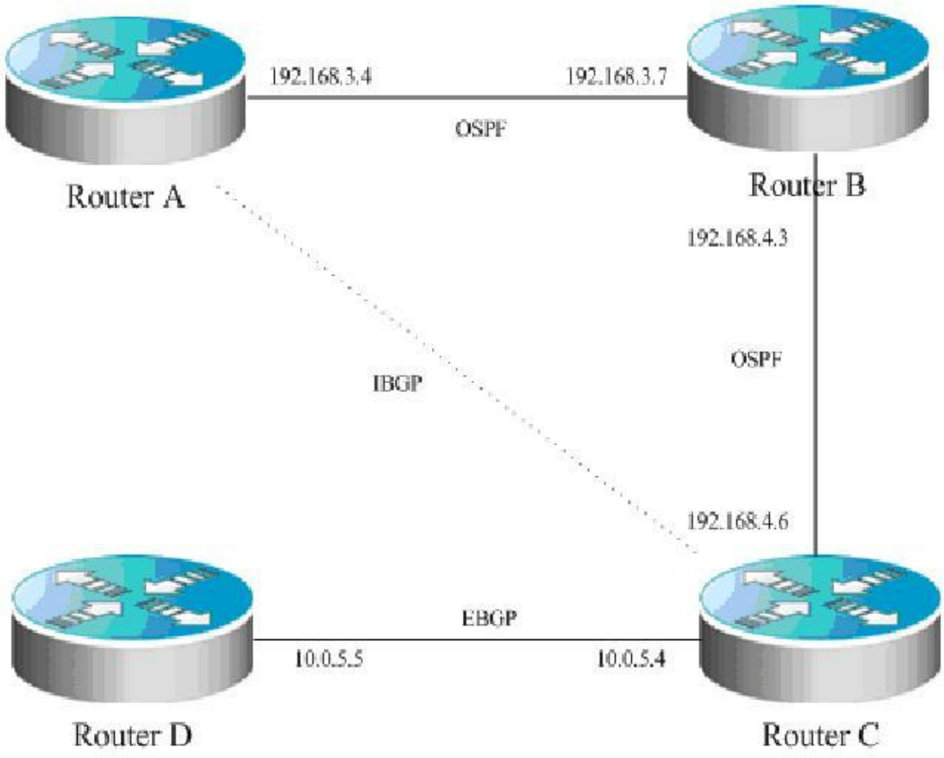
```
!
router bgp 200
 neighbor 192.168.5.2 remote-as 100
```

Configuring BGP Synchronization

Use the **synchronization** command to configure synchronization in BGP routing configuration mode, and use the **no synchronization** command to cancel the configured synchronization.

The following example shows the function of synchronization. The following figure illustrates the relationship between devices and the assignment of IP addresses:

Figure 4



In the figure, route p in router A is sent to router C based on the IBGP adjacency. If router C is configured with BGP synchronization, it is necessary for the router to wait for the IGP (this example uses the OSPF protocol) to receive the same routing information p, and send the route p to the EBGP neighbor, router D. If router C is configured asynchronously, it is not necessary for the BGP to wait for the IGP to receive the route p, and send the route p to the EBGP neighbor router D.

Configuring Neighbors to Use as-Path Filter

Configure the **as-path access-list** command for filtering first in configuration mode. Enter BGP route configuration mode after configuration, and use the **neighbor filter-list** command to apply the configured as-path access list among the BGP neighbors to filter AS paths.

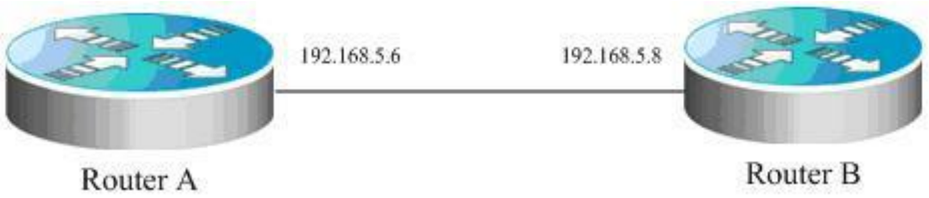
The configurations are detailed below:

```
router bgp 200
neighbor 193.1.12.10 remote-as 100
neighbor 193.1.12.10 filter-list 2 out
neighbor 193.1.12.10 filter-list 3 in
ip as-path access-list 2 permit _200$
ip as-path access-list 2 permit ^100$
ip as-path access-list 3 deny _690$
ip as-path access-list 3 permit .*
```

This configuration indicates that only the routes permitted by the **as-path access-list** *2* can be advertised to the neighbor 193.1.12.10. The advertised routes from the neighbor 193.1.12.10 can be received only when they are permitted by the **as-path access-list** *3.*

The following figure provides a configuration eaxmple that shows the relationship and IP addresses of devices:

Figure 5



Do AS path-based filter on Router A.

The following example shows the configuraitons of different devices:

Router A configuration:

```
!
ip as-path access-list 4 deny ^300_
ip as-path access-list 4 permit .*
ip as-path access-list 5 deny ^450_65_
ip as-path access-list 5 permit .*
!
router bgp 100
 bgp log-neighbor-changes
 neighbor 192.168.5.8 remote-as 200
 neighbor 192.168.5.8 filter-list 5 in
 neighbor 192.168.5.8 filter-list 4 out
```

Router B configuration:

```
!
router bgp 200
 bgp log-neighbor-changes
 neighbor 192.168.5.6 remote-as 100
```

Configuring Route Aggregation

Use the **aggregate-address** command to configure an aggregated route in route configuration mode. When any route falls within the configured range, this aggregated route will become active.

The configuration is detailed as follows:

```
router bgp 100
aggregate-address 193.0.0.0 255.0.0.0
```

Configure one aggregate route:

```
router bgp 100
aggregate-address 193.0.0.0 255.0.0.0 as-set
```

The **as-path** segment of the aggregated route is an collection of **ASs**:

```
router bgp 100
aggregate-address 193.0.0.0 255.0.0.0 summary-only
```

The aggregated route is not advertised.

Configuring Confederation

When configuring a confederatin, you need to use the **bgp confederation identifier** command to configure the AS number for external connection, and use the **bgp confederation peers** command to configure confederation members.

The configuration is detailed as follows:

```
router bgp 6003
bgp confederation identifier 666
bgp confederation peers 6001 6002
neighbor 171.69.232.57 remote-as 6001
neighbor 171.69.232.55 remote-as 6002
neighbor 200.200.200.200 remote-as 701
```

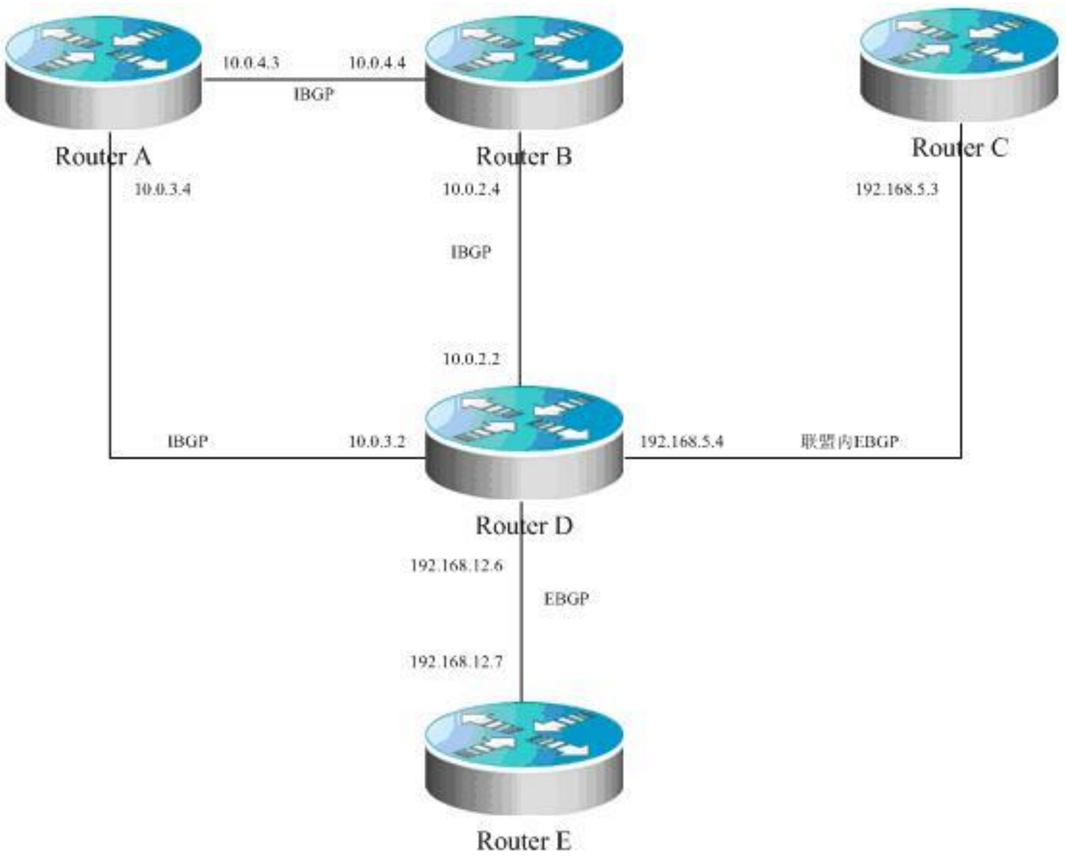
The configuration of peer 200.200.200.200 outside the confederation is shown as follows:

```
router bgp 701
neighbor 171.69.232.56 remote-as 666
neighbor 200,200,200,205 remote-as 701
```

For the configuration, the first device is in the confederation, while the second device is outside the confederation. Therefore, they are EBGP neighbors.

The following example shows their relastionship and IP addresses:

Figure 6



The following example shows the configurations of different devices:

Router A configuration:

```
!
router bgp 65530
 bgp confederation identifier 100
 bgp confederation peers 65531
 bgp log-neighbor-changes
 neighbor 10.0.3.2 remote-as 65530
 neighbor 10.0.4.4 remote-as 65530
```

Router B configuration:

```
!
router bgp 65530
 bgp confederation identifier 100
 bgp log-neighbor-changes
 neighbor 192.168.5.4 remote-as 65530
```

Router C configuration

```
!
router bgp 65531
 bgp confederation identifier 100
 bgp confederation peers 65530
 bgp log-neighbor-changes
 neighbor 10.0.3.2 remote-as 65530
 neighbor 10.0.4.4 remote-as 65530
```

Router D configuration:

```
!
router bgp 65530
 bgp confederation identifier 100
 bgp confederation peers 65531
 bgp log-neighbor-changes
 neighbor 10.0.2.4 remote-as 65530
 neighbor 10.0.3.4 remote-as 65530
 neighbor 192.168.5.3 remote-as 65531
 neighbor 192.168.12.7 remote-as 200
```

Router E configuration:

```
!
router bgp 200
 bgp log-neighbor-changes
 neighbor 192.168.12.6 remote-as 100
```

Configuring Route Reflector

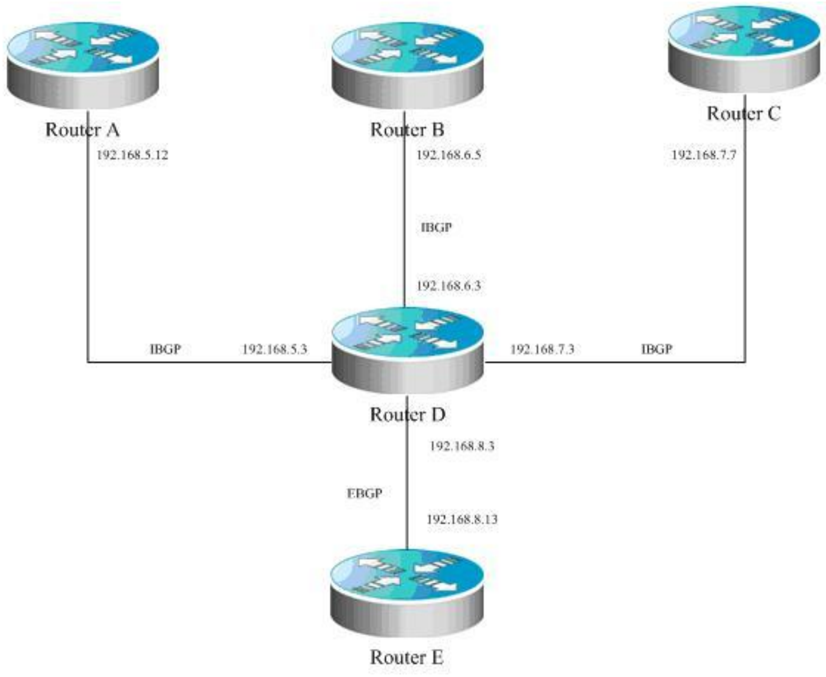
When a route reflector is configured, use the **bgp client-to-client reflection** command to enable the route reflection function on the device. If there are more than one route reflector within one cluster, use the **bgp cluster-id** command to configure the cluster ID of the reflector, and use the **neighbor route-reflector-client** command to add the peer to the client for route reflection.

The configuration is detailed as follows:

```
router bgp 601
bgp cluster-id 200.200.200.200
neighbor 171.69.232.56 remote-as 601
neighbor 200,200,200,205 remote-as 701
neighbor 171.69.232.56 route-reflector-client
```

The following example shows the relastionship between and IP addresses of the devices:

Figure 7



In this example, Router D is a route reflector. The following section shows the configurations of different devices:

Router A configuration:

```
!
router bgp 100
 bgp log-neighbor-changes
 neighbor 192.168.5.3 remote-as 100
 neighbor 192.168.5.3 description route-reflector server
```

Router B configuration:

```
!
router bgp 100
 bgp log-neighbor-changes
 neighbor 192.168.6.3 remote-as 100
 neighbor 192.168.6.3 description route-reflector server
```

Router C configuration:

```
!
router bgp 100
 bgp log-neighbor-changes
 neighbor 192.168.7.3 remote-as 100
 neighbor 192.168.7.3 description not the route-reflector server
```

Router D Configuration:

```
!
router bgp 100
```

```
 bgp log-neighbor-changes
 neighbor 192.168.5.12 remote-as 100
 neighbor 192.168.5.12 description route-reflector client
 neighbor 192.168.5.12 route-reflector-client
 neighbor 192.168.6.5 remote-as 100
 neighbor 192.168.6.5 description route-reflector client
 neighbor 192.168.6.5 route-reflector-client
 neighbor 192.168.7.7 remote-as 100
 neighbor 192.168.7.7 description not the route-reflector client
 neighbor 192.168.8.13 remote-as 200
```

Router E configuration:

```
!
router bgp 500
 bgp log-neighbor-changes
 neighbor 192.168.8.3 remote-as 100
```

Configuring Peergroup

This section uses the configuration of **peergroup** for IBGP and EBGP as an example.

Configuring IBGP peergroup

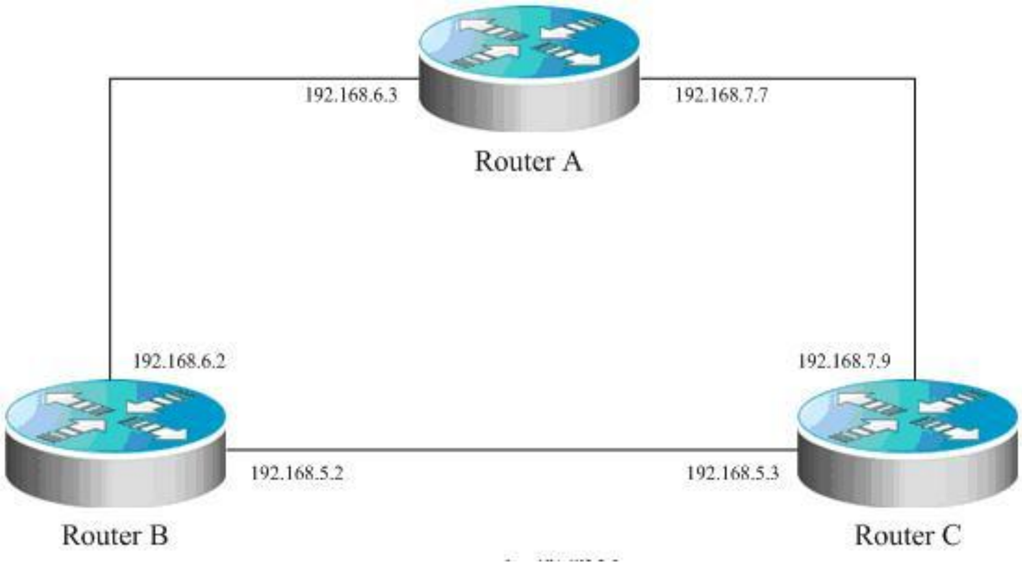
Use the **neighbor** *internal* **peer-group** command to create a peer group named *internal*, and configure a remote AS, and other options for the peer group. Use the **neighbor** *A.B.C.D* **peer-group** *internal* command to add peers A.B.C.D into the peer group.

The configuration commands are described as follows:

```
router bgp 100
neighbor internal peer-group
neighbor internal remote-as 100
neighbor internal update-source loopback 0
neighbor internal route-map set-med out
neighbor internal filter-list 1 out
neighbor internal filter-list 2 in
neighbor 171.69.232.53 peer-group internal
neighbor 171.69.232.54 peer-group internal
neighbor 171.69.232.55 peer-group internal
neighbor 171.69.232.55 filter-list 3 in
```

The following example shows the relastionship between and IP addresses of the devices:

Figure 8



Router A configuration

```
!
router bgp 100
 bgp log-neighbor-changes
 neighbor ibgp-group peer-group
 neighbor ibgp-group description peer in the same as
 neighbor 192.168.6.2 remote-as 100
 neighbor 192.168.6.2 peer-group ibgp-group
 neighbor 192.168.6.2 description one peer in the ibgp-group
 neighbor 192.168.7.9 remote-as 100
 neighbor 192.168.7.9 peer-group ibgp-group
```

Router B configuration:

```
!
router bgp 100
 bgp log-neighbor-changes
 neighbor ibgp-peer peer-group
 neighbor ibgp-peer remote-as 100
 neighbor ibgp-peer route-map ibgp-rmap out
 neighbor 192.168.5.3 peer-group ibgp-peer
 neighbor 192.168.5.3 route-map set-localpref in
 neighbor 192.168.6.3 peer-group ibgp-peer
```

Router C configuration:

```
!
router bgp 100
 bgp log-neighbor-changes
 neighbor ibgp-group peer-group
 neighbor 192.168.5.2 remote-as 100
 neighbor 192.168.5.2 peer-group ibgp-group
 neighbor 192.168.7.7 remote-as 100
 neighbor 192.168.7.7 peer-group ibgp-group
```

Configuring EBGP peergroup

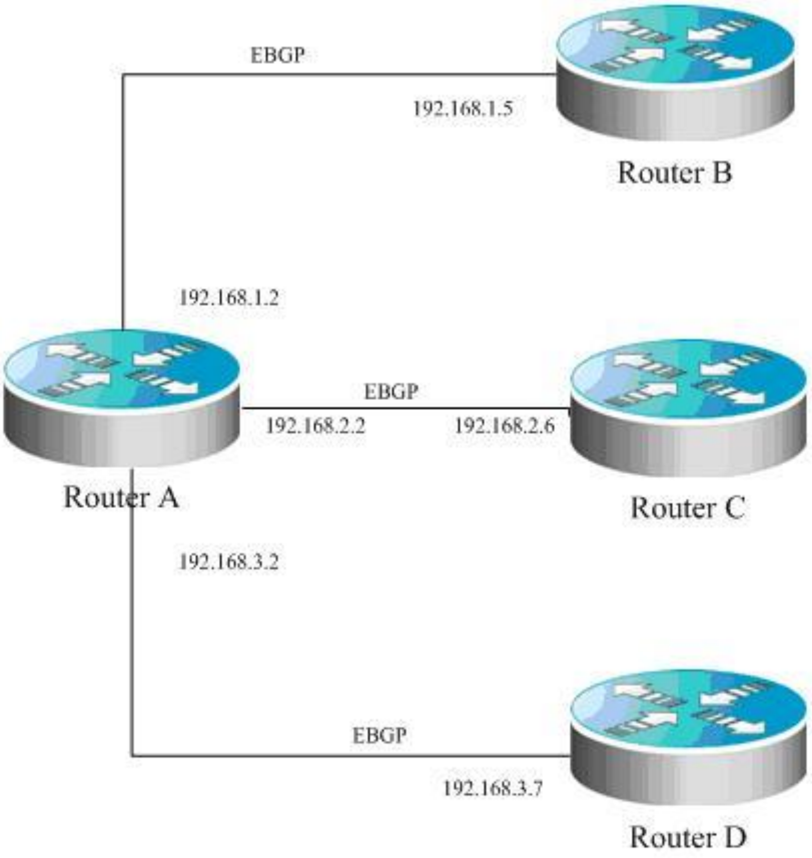
Use the **neighbor** *A.B.C.D* **remote-as** *num* command to configure an EBGP peer. Use the **neighbor** *external* **peer-group** command to create a peer group named **external**, and apply the **neighbor** *A.B.C.D* **peer-group** *external* command to add the peers A.B.C.D into the peer group *external*.

Here is an example of the specific configuration:

```
router bgp 100
neighbor external-peers peer-group
neighbor external-peers route-map set-metric out
neighbor external-peers filter-list 99 out
neighbor external-peers filter-list 101 in
neighbor 171.69.232.90 remote-as 200
neighbor 171.69.232.90 peer-group external-peers
neighbor 171.69.232.100 remote-as 300
neighbor 171.69.232.100 peer-group external-peers
neighbor 171.69.232.110 remote-as 400
neighbor 171.69.232.110 peer-group external-peers
neighbor 171.69.232.110 filter-list 400 in
```

The following figure shows the configuration of peer-group:

Figure 9



The figure illustrates the relationship between devices and the assignment of IP address.

Router A configuration:

```
!
router bgp 100
 bgp log-neighbor-changes
 neighbor ebgp-group peer-group
 neighbor ebgp-group distribute-list 2 in
 neighbor ebgp-group route-map set-med out
 neighbor 192.168.1.5 remote-as 200
 neighbor 192.168.1.5 peer-group ebgp-group
 neighbor 192.168.2.6 remote-as 300
 neighbor 192.168.2.6 peer-group ebgp-group
 neighbor 192.168.2.6 distribute-list 3 in
 neighbor 192.168.3.7 remote-as 400
 neighbor 192.168.3.7 peer-group ebgp-group
!
```

Router B configuration:

```
!
router bgp 200
```

```
 bgp log-neighbor-changes
 neighbor 192.168.1.2 remote-as 100
!
```

Router C configuration:

```
!
router bgp 300
 bgp log-neighbor-changes
 neighbor 192.168.2.2 remote-as 100
!
```

Router D configuration:

```
!
router bgp 400
 bgp log-neighbor-changes
 neighbor 192.168.3.2 remote-as 100
!
```

Configuring TCP MD5

Use the CLI command **neighbor password** to configure TCP MD5 for the BGP connection in BGP configuration mode.

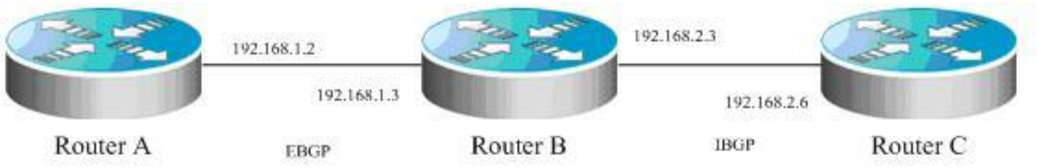
The configuration format is shown as follows:

```
router bgp 100
neighbor 171.69.232.54 remote-as 110
neighbor 171.69.232.54 password peerpassword
```

Configure the *password* of peer 171.69.232.54 as *peerpassword.*

The following figure shows the configuration of MD5 and IP addresses on different devices：

Figure 10



The AS of router A is 100, and the AS of router B and router C is 200. Router A establishes EBGP adjacency with router B and uses EBGP as the MD5 password. Router B establishes IBGP adjacency with router C and uses IBGP as the MD5 password.

router A configuration:

```
!
router bgp 100
 bgp log-neighbor-changes
 neighbor 192.168.1.3 remote-as 200
 eighbor 192.168.1.3 password ebgp
!
```

Router B configuration:

```
!
router bgp 200
 bgp log-neighbor-changes
 neighbor 192.168.1.2 remote-as 100
 neighbor 192.168.1.2 password ebgp
 neighbor 192.168.2.6 remote-as 200
 neighbor 192.168.2.6 password ibgp
!
```

Router C configuration:

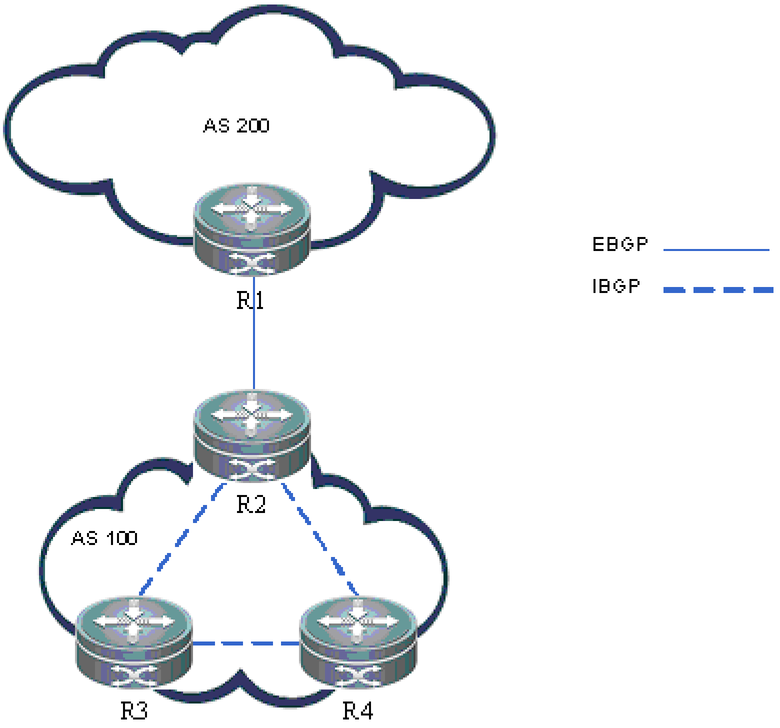
```
!
router bgp 200
 bgp log-neighbor-changes
 neighbor 192.168.2.3 remote-as 200
 neighbor 192.168.2.3 password ibgp
!
```

Configuring BGP GR

Networking Requirements

As shown in the following figure, R2 is the border device of AS100 and AS200. R1 is the access device of AS200. In AS100, R2, R3 run OSPF to offer IBGP connection for the BGP protocol. At the same time, IBGP connections are established between them. R2 establishes an EBGP conneciton with R1. R2, as the border device connecting AS100 and AS200, must be more reliable. R2 is configured to support dual-system redundant bakcup for continuous forwarding and graceful restart of routing protocols (OSPF and BGP in this example). The graceful restart of routing protocols involves adjacent devices. Hence, R1, 3 and R4 need to support the BGP GR capability, and R3 and R4 need to support the GR Helper of OSPF to support the OSPF GR capabiltiy. In this way, when one engine of R2 fails, the transmission of data is not interrupted and therefore reliability is enhanced.

Figure 11 BGP GR configuration example



Confiugration Precautions:

Before configuration, ensure that R2 can serve as the GR Restarter for graceful restart and the software on all devices support OSPF GR and BRP GR capability. If not, continuous data forwarding cannot be peformed when the backup engine takes over the work of the master engine in case of failures. Meanwhile, the BGP protocol depends on the BGP connection from OSPF. Hence, both the BGP and OSPF protocols should have GR enabled. Therefore, R2 must support OSPF GR.

14) R2 enables dual-eningle redundnat hot backup;

15) The software of all devices support OSPF GR and BGP GR capability

16) OSPF GR is enabled on R2

17) BGP GR is enabled on R2

18) BGP GR is enabled on neighbors to support the GR Helper of BGP

19) All BGP connections restart on R2 to negotitate GR capabitliy

Configuration Steps

20) Ensure that R2 enables dual-eningle redundnat hot backup

21) Ensure that the software of all devices supports OSPF GR and BGP GR capabilities.

Check that these devices support the configuration commands of BGP GR and OSPF GR. For details, refer to Step 3 and Step 4.

22) Enables OSPF GR on R2

```
Ruijie(config)# router ospf 1
Ruijie(config-router)# graceful-restart
```

23) Enable BGP GR on R2

```
Ruijie(config)# router bgp 100
Ruijie(config-router)# graceful-restart
```

24) Enable BGP GR on neighbors to support the GR Helper of BGP

```
Ruijie(config)# router bgp 100
Ruijie(config-router)# bgp graceful-restart
```

For BGP GR negotiation, both sides of the BGP connection must enable BGP GR. Hence, R2 needs to negotiate with its neighbors which serve as the GR Helper to assist BGP GR.

25) Restart all BGP connections on R2 to neotitate GR capabitliy

You must manually restart the BGP connection for GR capability renegotiation, because the **bgp graceful-restart** command cannot take effect immediately.

```
Ruijie# clear ip bgp *
```

Configuration Check

For R2 to enable continuous data forwarding during engine handover, check the negotiation of BGP GR and OSPF GR configration.

26) Ensure that BGP GR can negitate with all neighbors.

```
 Ruijie# show ip bgp neighbors
BGP neighbor is 192.168.195.183, remote AS 200, local AS 100, external link
Using BFD to detect fast fallover - BFD session state up
  BGP version 4, remote router ID 10.0.0.1
  BGP state = Established, up for 00:06:37
  Last read 00:06:37, hold time is 180, keepalive interval is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received (old and new)
Address family IPv4 Unicast: advertised and received
Graceful restart: advertised and received
  Remote Restart timer is 120 seconds
  Address families preserved by peer:
    None
```

Graceful restart: advertised and received means BGP GR negotiation of the BGP connection is successful. Ensure that BGP GR can negotiate with all BGP connections.

27) Ensure that OSPF GR is enabled on R2.

```
Ruijie# show ip ospf
Routing Process "ospf 1" with ID 10.0.0.2
Process uptime is 4 minutes
Process bound to VRF default
Conforms to RFC2328, and RFC1583Compatibility flag isenabled
Supports only single TOS(TOS0) routes
Supports opaque LSA
This router is an ASBR (injecting external routing information)
SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
LsaGroupPacing: 240 secs
Number of incomming current DD exchange neighbors 0/5
Number of outgoing current DD exchange neighbors 0/5
Number of external LSA 4. Checksum 0x0278E0
Number of opaque AS LSA 0. Checksum 0x000000
Number of non-default external LSA 4
External LSA database is unlimited.
Number of LSA originated 6
Number of LSA received 2
Log Neighbor Adjency Changes : Enabled
Graceful-restart enabled
Graceful-restart helper support enabled
Number of areas attached to this router: 1
Area 0 (BACKBONE)
```

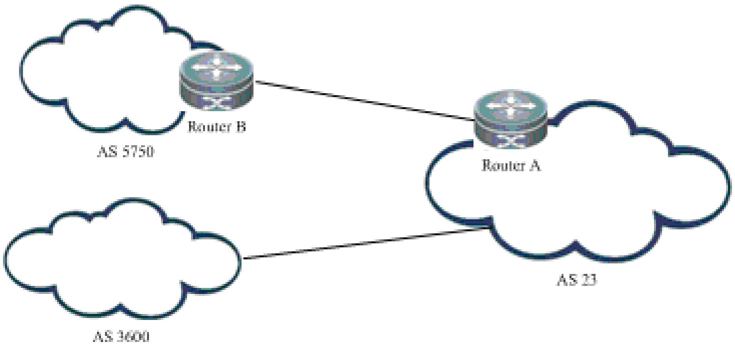
Graceful restart enabled means OSPF GR is enabled.

Configuring BGP Local AS

Networking requirements

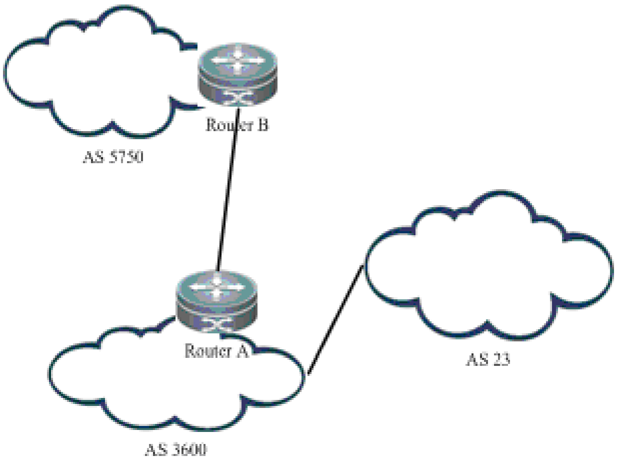
As shown in the following figure, Router A and its home network are located in AS 23, which connects AS 3600 through EBGP. The routing information of AS 5750 is transmitted to AS 3600 via AS 23.

Figure 12 Logical topology before AS migration



You must migrate Router A and its home network to AS 3600.

Figure 13 Logical topology after AS migration

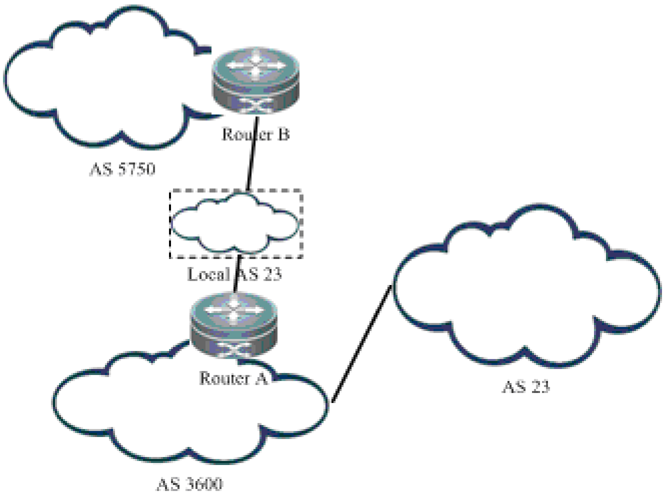


AS 23 and AS 3600 belong to one management domain. The configurations of these two ASs are modified after negotiation. At this point, Router A configures AS 3600 as the AS of the BGP protocol. In this case, you need to maintain the BGP connection between Router A and Router B, and modify related peer configuration on Router B of AS 5750. Sometimes Router B may not modify the configuration immediately. As a result, Router B cannot establish the BGP connection with Router A. On Router A, you can configure local AS for Router B to establish a BGP connection between them wihtout affecting the transmisison and calculation of routes.

Networking Topology

The following figure illustrates how to configure local AS for Router B.

Figure 14 Configuration of local AS



After configuration, a virtual AS 23 is set up between Router A and Router B. Router B considers that it is directly connected to AS 23 and can transmit routes to AS 23. This removes the need to modify Router B’s configuration. When AS 3600 reaches an agreement with AS 5750 in terms of management, Router B can modify the remote AS of Router A as AS 3600 and Router A deletes the corresponding local AS for migration of network in different ASs.

Configuration Steps

28) Enter BGP configuration mode

```
Ruijie-A(config)# router bgp 3600
```

29) Configures local AS for the peer

```
Ruijie-A(config-router)# neighbor 57.50.1.1 local-as 23 no-prepend replace-as dual-as
```

30) Delete local AS after Router B modifies its configuration

```
Ruijie-A(config-router)#no neighbor 57.50.1.1 local-as
```

Configuration Check

```
Use the show ip bgp neighbors command to verify the local AS used by a neighbor to establish
a BGP connection as follows:
Ruijie-A#show ip bgp neighbors 57.50.1.1
BGP neighbor is 57.50.1.1,  remote AS 5750, local AS 23(using Peer's Local AS, no-prepend,
replace-as, dual-as), external link
  BGP version 4, remote router ID 0.0.0.0
  BGP state = Idle
  Last read        , hold time is 180, keepalive interval is 60 seconds
  Received 0 messages, 0 notifications, 0 in queue
    open message:0 update message:0 keepalive message:0
    refresh message:0 dynamic cap:0 notifications:0
  Sent 0 messages, 0 notifications, 0 in queue
```

Detailed Configuration:

Router A configuration

```
router bgp 3600
neighbor 57.50.1.1 remote-as 5750
neighbor 57.50.1.1 local-as 23 no-prepend replace-as dual-as
neighbor 57.50.1.1 update-source loopback 0
neighbor 57.50.1.1 ebgp-multihop 255
```

Router B configuration

```
router bgp 5750
neighbor 36.0.1.1 remote-as 23
neighbor 36.0.1.1 update-source loopback 0
neighbor 36.0.1.1 ebgp-multihop 255
```

Configuring BGP MCE

About BGP MCE

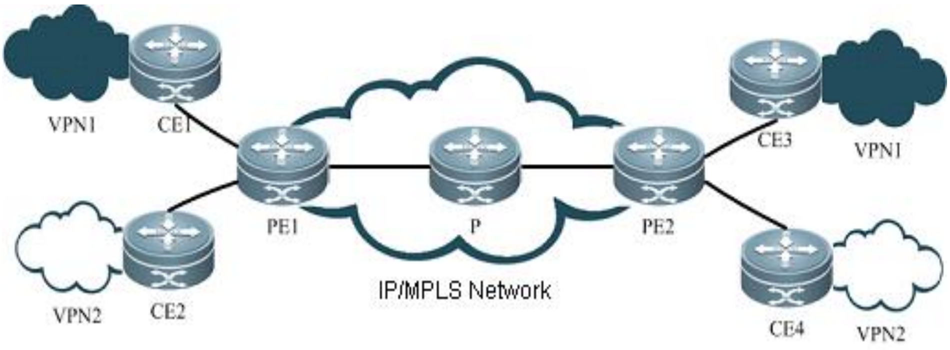
MCE Overview

MCE refers to Multi-CE. MCE enabled network devices can function as CEs of multiple VPN instances in a BGP/MPLS VPN network. This helps reduce the need for additional network equipment.

Working Principle of BGP MCE

With BGP/MPLS VPN, private network data can be securely transmitted in a public network through tunnels. However, in a typical BGP/MPLS VPN network, each VPN is connected to the PE through a CE, as shown in Figure 15:

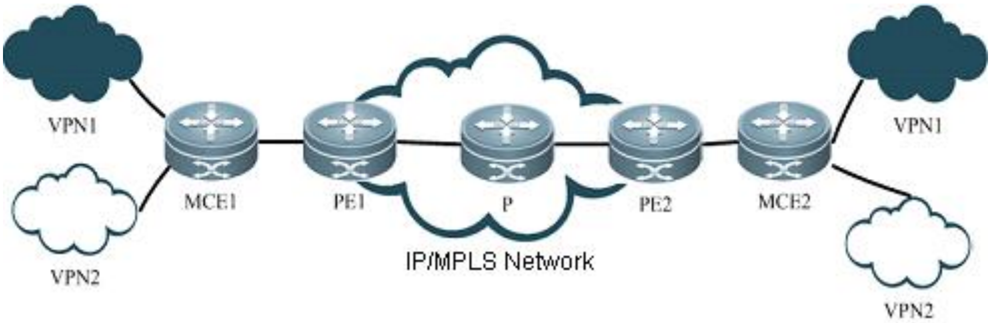
Figure 15 BGP/MPLS VPN network



As users’ demand surges for service segmentation and security, a private network may be divided into multiple VPNs, and the users of different VPNs are usually isolated from each other. As a result, equipment and maintenance costs may be increased by assigning a CE for each VPN, while data security cannot be guaranteed by sharing one CE and using the same routing entry among multiple VPNs. MCE can balance data security and networking cost. By binding the VLAN interfaces of a CE device to the VPNs, you can create and maintain a routing table for each of the VPNs (Multi-VRF). In this way, packets of different VPNs in the private network can be isolated. Moreover, the PE enables the routes of each VPN to be advertised to the corresponding remote PE. As such, packets of each VPN can be transmitted securely through the public network.

The following example shows how the MCE maintains routing entries of multiple VPNs and how the MCE exchanges VPN routes with PEs.

Figure 16 MCE functions



As shown in Figure 16, two VPN sites on the left side (VPN1 and VPN2) are connected to the MPLS backbone through an MCE device. Users of VPN1 and VPN2 need to establish VPN tunnels with remote VPN1 and VPN2 users. MCE enables routing tables to be created for VPN1 and VPN2 individually on the MCE device. VLAN-interface 2 can be bound to VPN1, and VLAN-interface 3 can be bound to VPN 2. When receiving routing information, MCE determines the source of information based on the number of the interface receiving the information and then maintains the corresponding VPN routing table. Meanwhile, you need to bind the MCE-connecting interfaces on PE1 to the VPNs in the same way as those on the MCE device. The MCE device is connected to PE1 through a trunk, which permits packets of VLAN2 and VLAN3 carrying VLAN tags. In this way, PE1 can determine the home VPN of a received packet according to the VLAN tag and passes the packet to the corresponding tunnel.

How does MCE device accurately transmit private routing information of multiple VPN instances to PEs? This involves two steps: routing information exchange between MCE and VPN site, and between MCE and PE. There are several ways to exchange routing information, such as static route, RIP, OSPF, ISIS and BGP. If BGP routing protocol is used to exchange routing information, BGP MCE applies. Specifically, BGP MCE allows BGP protocol to support VRF and enable BGP routing information exchange under VRF. We need to configure the BGP peer for each VRF instance on MCE and introduce IGP routing information of corresponding VPN. As each VPN is generally in different ASs, EBGP is therefore used to advertise routes.

Protocol Specification

NA.

Default Configurations

The following table describes the default configurations of BGP MCE.

Function	Default setting
VRF instance	No VRF instance is created by default.
BGP-VRF binding	No BGP-VRF binding by default

The following products support BGP MCE: the RSR30, RSR50 and RSR50E series of routers.

Configuring BGP MCE

Configuring VRF Instance and Route-related Attributes

Before configuring the BGP MCE, VRF instance and route-related attributes must be configured first:

Command	Function
Ruijie # **configure terminal**	Enters global configuration mode
Ruijie(config)# **ip vrf** *VRF1*	Creates a VRF named VRF1 and enters VRF mode.
Ruijie(config-vrf)# **rd** *rd-value*	Configures VRF RD value, which is identified using XX:XX format, such as RD 1:100. 1 refers to the AS ID of backbone network, while 100 is a user-defined numerical value.
Ruijie(config-vrf)# **route-target both** | **export** | **import**} *rt-value*	Configures the route export and import RT attribute of VRF.
Ruijie(config-vrf)# {**export** | **import**} **map** *map*	Configures the route map for import and export routes to support policy-based filtering of import and export routes.
Ruijie(config-vrf)# **exit**	Exits VRF mode and enters global configuration mode
Ruijie(config)# **interface vlan** ***2***	Enters VLAN 2 interface configuration mode.
Ruijie(config-if)# **ip vrf forwarding** *VRF1*	Associates interface with VRF instance of VRF2
Ruijie(config-if)# **ip address 172.16.25.18 255.255.255.0**	Configures IP address for VLAN 2
Ruijie(config-if)# **end**	Returns to privileged EXEC mode
Ruijie # **show running-config**	Verifies the configurations
Ruijie # **write**	(Optional) Saves configurations.

Configuring BGP Route Exchange between MCE and VPN Site

To use the BGP protocol to exchange routing information between MCE and VPN sites, you need to bind BGP to the corresponding VRF instance on MCE, and configure site device as EBGP neighbor, as shown below:

Command	Function
Ruijie # **configure terminal**	Enters global configuration mode
Ruijie(config)# **router bgp** ***23***	Enables the BGP protocol and enters BGP routing process mode
Ruijie(config-router)# **address-family ipv4 vrf** *VRF1*	Enters the IPv4 address family configuration mode of VRF1
Ruijie(config-router-af)# **neighbor** 172.16.25.57 **remote-as** 65531	Configures EBGP neighbor and learns routing information advertised by VPN site through BGP.
Ruijie(config-router-af)# **redistribute** *ospf 1*	Introduces the routing information of remote VPN as advertised by PE. We assume that MCE and PE exchange routing information through OSPF protocol.
Ruijie(config- router-af))# **end**	Returns to privileged EXEC mode.
Ruijie # **show running-config**	Verifies the configurations.
Ruijie # **write**	(Optional) Saves configurations.

BGP protocol must also be enabled on CE devices at a VPN site, allowing the site to exchange routing information with MCE devices through BGP protocol.

Configuring BGP Route Exchange between MCE and PE

To use the BGP protocol to exchange routing information between MCE and PE, bind BGP to the corresponding VRF instance on MCE, and configure PE device as EBGP neighbor, as shown below:

Command	Function
Ruijie # **configure terminal**	Enters global configuration mode.
Ruijie(config)# **router bgp *23***	Enables BGP protocol and enter BGP routing mode.
Ruijie(config-router)# **address-family ipv4 vrf** *VRF1*	Enters IPv4 address family configuration mode of VRF1.
Ruijie(config-router-af)# **neighbor** 172.16.25.157 **remote-as** 65532	Configures EBGP neighbor and study the routing information advertised by PE through BGP.
Ruijie(config-router-af)# **redistribute** *ospf 1*	Introduces the routing information of local VPN. We assume that MCE and local VPN site exchange routing information through OSPF protocol.
Ruijie(config- router-af))# **end**	Returns to privileged EXEC mode.
Ruijie # **show running-config**	Verifies the configurations.
Ruijie # **write**	(Optional) Saves configurations.

BGP protocol must also be enabled on the PE device, and MCE must be configured as EBGP neighbor, allowing PE to exchange routing information with the MCE device through BGP protocol.

Displaying Configurations

The "show" commands used in BGP MCE are similar to the "show" commands used in ordinary BGP. The following information is displayed: neighbor state, routing information and neighbor summary.

Command	Function
Ruijie # **show ip vrf**	Displays all VRF summary information on the device.
Ruijie # **show ip vrf detail [*VRF1*]**	Displays the detailed configurations of all VRFs or a specified VRF.
Ruijie # **show ip vrf interfaces** [*VRF1*]	Displays the interface binding information and state of all VRFs or a specified VRF.
Ruijie# show **ip bgp vrf** *VRF1* [**summary**| **neighbors**| *A.B.C.D*]	Displays the summary information, detailed information, specific routing information, and all routing information of BGP neighbor under VRF1. Similar to those "show" commands used in ordinary BGP, other sub-commands are not discussed herein.
Ruijie# **show bgp vpnv4 unicast** [**all** | **rd** *rd* | **vrf** *vrf-name*] [**neighbors** | **summary** | *A.B.C.D*]	This command is similar to the above command, but the routes displayed are different: "all" will display all vpn routes, "rd" will display vpn routes with a specified RD value, and "vrf" will display vpn routes under a specified VRF.

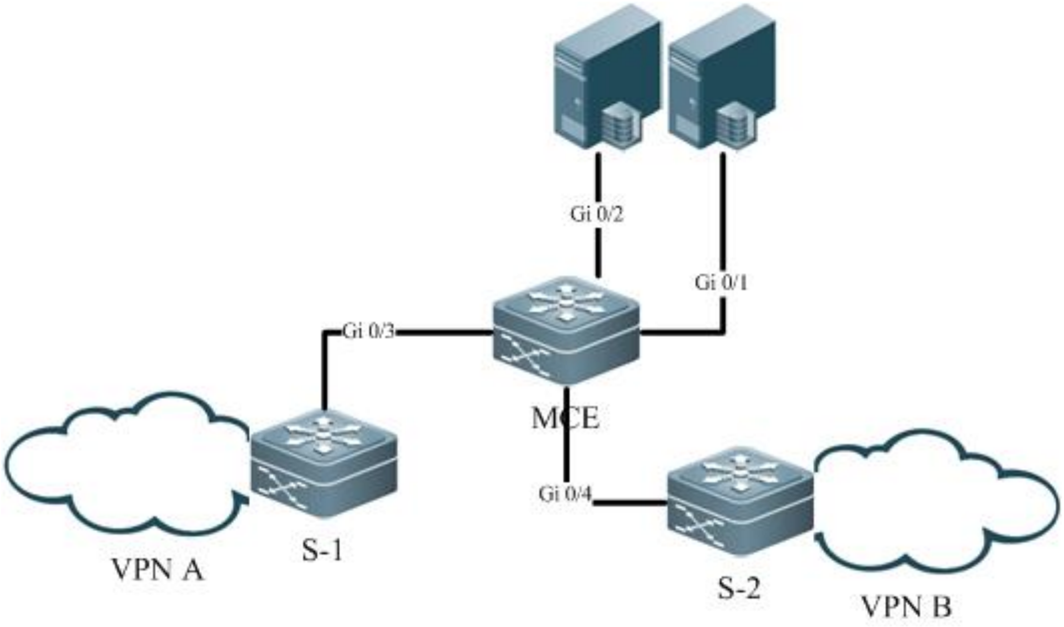
Typical BGP MCE Configuration Examples

Networking Requirements

A company needs to isolate the networks of two subsidiaries: A and B, and expects both subsidiaries to access the resource servers at the same time. OSPF protocol operates on the networks of subsidiary A and subsidiary B. An MCE device is used to isolate A and B. The device is directly connected with multiple resource servers.

Network Topology

Figure 17 Network topology of BGP MCE



S-1 is the convergence device on the network of subsidiary A; S-2 is the convergence device on the network of subsidiary B. S-1 and S-2 are both connected with the MCE device, with connecting interfaces belonging to different VRF instances and running the OSPF routing protocol. Gi 0/1 and Gi 0/2 of the MCE is directly connected with resource servers, and belong to another separate VRF instance.

Configuration Tips

We assume that OSPF protocol is running normally on VPN A connected directly with S-1 and VPN B connected directly with S-2. The following configurations are related to the MCE device.

31) Configure VRF instances and associate with interfaces to allow network isolation;

32) Configure OSPF routing protocol and associate with VRF, so that MCE can learn the routes to respective subsidiaries;

33) Configure BGP routing protocol and import OSPF routes and directly connected routes, allowing route exchange between different VRFs;

34) Configure the import and export route attributes of VRF instance, so that both subsidiaries can access the resource servers;

Configuration Steps

35) Configures VRF instances and associate with interfaces;

# Create three VRF instances: VRF1, VRF2 and VRF3

```
Ruijie# config terminal
Ruijie(config)# ip vrf VRF1
Ruijie(config-vrf)# rd 100:1
Ruijie(config-vrf)# exit
Ruijie(config)# ip vrf VRF2
Ruijie(config-vrf)# rd 100:2
Ruijie(config-vrf)# exit
Ruijie(config)# ip vrf VRF3
Ruijie(config-vrf)# rd 100:3
Ruijie(config-vrf)# exit
```

# Bind Gi0/1 and Gi0/2 to VRF3, Gi0/3 to VRF1 and Gi0/4 to VRF2

```
Ruijie(config)#interface gigabitEthernet 0/1
Ruijie(config-GigabitEthernet 0/1)#ip vrf forwarding VRF3
Ruijie(config-GigabitEthernet 0/1)#ip address 10.1.1.1 255.255.255.0
Ruijie(config)#interface gigabitEthernet 0/2
Ruijie(config-GigabitEthernet 0/2)#ip vrf forwarding VRF3
Ruijie(config-GigabitEthernet 0/2)#ip address 10.1.2.2 255.255.255.0
Ruijie(config)#interface gigabitEthernet 0/3
Ruijie(config-GigabitEthernet 0/3)#ip vrf forwarding VRF1
Ruijie(config-GigabitEthernet 0/3)#ip address 172.16.25.18 255.255.255.0
Ruijie(config)#interface gigabitEthernet 0/4
Ruijie(config-GigabitEthernet 0/4)#ip vrf forwarding VRF2
Ruijie(config-GigabitEthernet 0/4)# ip address 192.168.25.18 255.255.255.0
```

36) Configure OSPF routing protocol and associate with VRF;

# Create OSPF 1 and OSPF 2 and associate with VRF1 and VRF2

```
Ruijie# config terminal
Ruijie(config)# router ospf 1 vrf VRF1
Ruijie(config-router)# network 172.16.25.0 0.0.0.255 area 0
Ruijie(config-router)# exit
Ruijie(config)# router ospf 2 vrf VRF2
Ruijie(config-router)# network 192.168.25.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

37) Configure BGP protocol and import routes

# Configure BGP protocol and import OSPF routes to BGP VRF instance

```
Ruijie# config terminal
Ruijie(config)# router bgp 100
Ruijie(config-router)# address-family ipv4 vrf VRF1
Ruijie(config-router-af)# redistribute ospf 1
Ruijie(config-router-af)# exit
Ruijie(config-router)# address-family ipv4 vrf VRF2
Ruijie(config-router-af)# redistribute ospf 2
```

# Import the directly connected routes of VRF3 to BGP VRF3 instance

```
Ruijie(config-router)# address-family ipv4 vrf VRF3
Ruijie(config-router-af)# redistribute connect
```

38) Configure the import and export route attributes of VRF instance

# Configure the import route attribute of VRF1 as 100:3 and export attribute as 100:1

```
Ruijie(config)# ip vrf VRF1
Ruijie(config-vrf)# route-target import 100:3
Ruijie(config-vrf)# route-target export 100:1
Ruijie(config-vrf)# exit
```

# Configure the import route attribute of VRF2 as 100:3 and export attribute as 100:2

```
Ruijie(config)# ip vrf VRF2
Ruijie(config-vrf)# route-target import 100:3
Ruijie(config-vrf)# route-target export 100:2
Ruijie(config-vrf)# exit
```

# Configure the import route attribute of VRF3 as 100:1 and 100:2 and export attribute as 100:3

```
Ruijie(config)# ip vrf VRF2
Ruijie(config-vrf)# route-target import 100:1 100:2
Ruijie(config-vrf)# route-target export 100:3
Ruijie(config-vrf)# exit
```

Verification

Execute the following steps to verify configurations:

39) Verify the state of interfaces bound to VRF. Execute the **show ip vrf interface** command to verify interface binding information and the interface state.

40)

```
Ruijie#sh ip vrf interfaces
Interface           IP-Address     VRF                 Protocol
GigabitEthernet 0/1  10.1.1.1      VRF3               up
GigabitEthernet 0/2  10.1.2.1      VRF3               up
GigabitEthernet 0/3  172.16.25.18  VRF1               up
GigabitEthernet 0/4  192.168.25.18 VRF2               up
```

41) Verify whether OSPF protocol bindings are correct and whether OSPF protocol runs normally. Execute the **show ip ospf** command to verify whether the OSPF instance is properly bound to VRF;

42) Verify whether the routes imported by BGP instance are correct. Execute the **show ip bgp vrf** or **show bgp vpnv4 unicast** command to verify whether the imported routes are correct, as shown below:

```
Ruijie#sh ip bgp vrf VRF3
BGP table version is 1, local router ID is 10.14.219.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network        Next Hop          Metric  LocPrf     Weight Path
*> 10.1.1.0/24   0.0.0.0             0       32768      ?
*> 10.1.2.0/24   0.0.0.0            0       32768     ?
Total number of prefixes 2
```

43) Execute the **show ip bgp vrf** command to verify whether routes of other VRFs have been properly imported to local VRF. Execute the **show ip route vrf** command to verify whether routes are correct.

```
Ruijie#sh ip bgp vrf VRF3
BGP table version is 1, local router ID is 10.14.219.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network         Next Hop          Metric  LocPrf     Weight Path
*> 10.1.1.0/24   0.0.0.0            0       32768     ?
*> 10.1.2.0/24   0.0.0.0            0       32768     ?
*> 172.16.22.0/24 0.0.0.0            0       32768     ?
*> 172.16.23.0/24 0.0.0.0            0       32768     ?
*> 172.16.25.0/24 0.0.0.0            0       32768     ?
*> 192.168.22.0   0.0.0.0            0       32768     ?
*> 192.168.23.0   0.0.0.0            0       32768     ?
*> 192.168.25.0   0.0.0.0            0       32768     ?
Total number of prefixes 8
Ruijie#sh ip route vrf VRF1
Routing Table: VRF1
Codes:  C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default
Gateway of last resort is no set
C    10.1.1.0/24 is directly connected, GigabitEthernet 0/1
C    10.1.1.1/32 is local host.
C    10.1.2.0/24 is directly connected, GigabitEthernet 0/2
C    10.1.2.1/32 is local host.
B    172.16.22.0/24 [20/0] via 0.0.0.0, 00:10:46, GigabitEthernet 0/3
B    172.16.23.0/24 [20/0] via 0.0.0.0, 00:10:46, GigabitEthernet 0/3
B    172.16.25.0/24 [20/0] via 0.0.0.0, 00:10:46, GigabitEthernet 0/3
B    192.168.22.0/24 [20/0] via 0.0.0.0, 00:10:46, GigabitEthernet 0/4
B    192.168.23.0/24 [20/0] via 0.0.0.0, 00:10:46, GigabitEthernet 0/4
B    192.168.25.0/24 [20/0] via 0.0.0.0, 00:10:46, GigabitEthernet 0/4
```

Configuring BGP 4-Octet AS

About 4-octet AS

Overview

A traditional AS number consists of two octets within the range of 1-65535. The AS number defined by RFC4893 consists of four octets falling within the range of 1-4294967295 to ease the burden of AS number resources. According to RFC5396, a 4-octet AS number supports two representation formats: asplain and asdot+. The two representations take the same format. Specifically, the 4-octet AS number will be represented using decimal value. The asdot+ representation contains ([high order 2 octets.] low order 2 octets). The high order 2 octets are not displayed if the value is 0. In other words, the AS number of 65536 in asplain format will be represented as 1.0 in asdot+ format. In addition, the AS number of 65534 in asplain format will be represented as 65534 in asdot+ format (without displaying the 0 value).

Working Principle

The 4-octet AS number requires a BGP connection between an old bgp speaker supporting only 2-octet AS number and a new bgp speaker supporting 4-octet AS number. If the autonomous system for the new bgp speaker uses a 4-octet AS number, the old bgp speaker must use the reserved AS number of 23456 to replace the 4-octet AS number of new bgp speaker while creating a neighbor. In the packets sent from the new bgp speaker to the old bgp speaker, 23456 will replace the 4-octet AS number in the domain of "My Autonomous System". Meanwhile, in the UPDAT packets sent to the old bgp speaker, 23456 will replace the 4-octet AS number found in AS-PATH and AGGREGATOR attributes. These packets also carry the true 4-octet AS number reserved in the optional transitive attributes of AS4-PATH and AS4-AGGREGATOR. Therefore, the true AS-PATH attribute and AGGREGATOR attribute can be restored when this route reaches the next new bgp speaker.

In other cases, the true AS number of peer side is directly used to create a neighbor.

Protocol Specification

RFC 4893

RFC 5396

Default Configurations

By default, BGP protocol is not enabled. After BGP protocol is enabled, the decimal value is used by default to represent 4-octet AS numbers.

Configuring BGP 4-octet AS

Configuring BGP Instance with 4-octet AS Number

Command	Function
Ruijie # **configure terminal**	Enters global configuration mode
Ruijie(config)# **router bgp 65538**	Enables BGP protocol and configure device AS number as 65538
Ruijie(config)# **router bgp 1.2**	Uses asdot+ format 1.2 to represent four-octet AS number of 65538
Ruijie(config-router)# **end**	Returns to privileged EXEC mode.
Ruijie # **write**	(Optional) Saves configurations.

Configuring the Display Format of 4-octet AS Number

By default, the asplain format is used to display a 4-octet AS number. You can also configure the display format as asdot+. Meanwhile, after changing the display format of a 4-octet AS number, the 4-octet AS number in regular expression will be matched using asdot+ format.

Command	Function
Ruijie # **configure terminal**	Enters global configuration mode
Ruijie(config)# **router bgp 65538**	Enables BGP protocol and configures the device AS number as 65538
Ruijie(config-router)# **bgp asnotation dot**	Use the asdot+ format to display 4-octet AS number, namely 1.2
Ruijie(config-router)# **end**	Returns to privileged EXEC mode.
Ruijie # **clear ip bgp** *	Resets BGP protocol for re-matching the regular expression.
Ruijie # **write**	(Optional) Saves configurations.

After executing the **bgp asnotation dot** command, you must execute the **clear ip bgp *** command to reset BGP protocol, so that the regular expression can be rematched.

Displaying Configurations

Execute the **show** command to view the configuration of the 4-octet AS number. This command is similar to the **show** command used in BGP mode.

Command	Function
Ruijie # **show ip bgp summary**	Displays the connection state of all BGP neighbors.

Typical BGP 4-Octet AS Configuration Examples

Interconnection between 4-octet AS and 2-octet AS

Networking Requirements

44) A BGP connection is established between the router supporting 2-octet AS number and the router supporting 4-octet AS number (using 2-octet AS number);

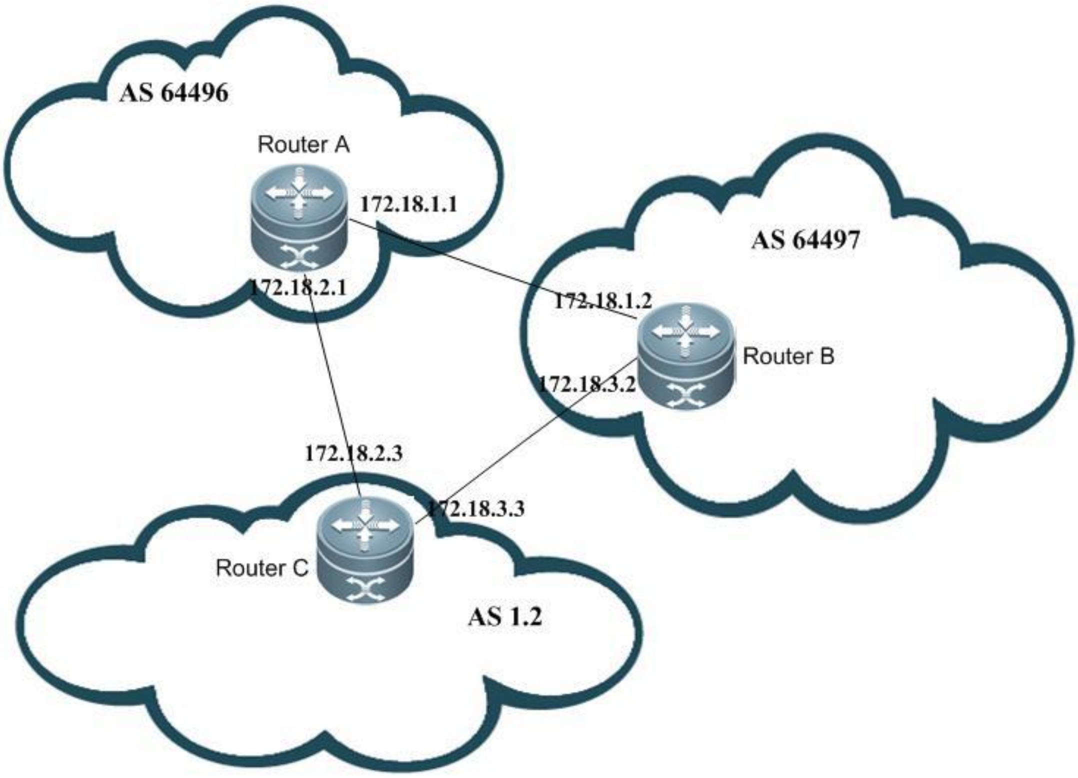
45) A BGP connection is established between the router supporting 2-octet AS number and the router supporting 4-octet AS number (using 4-octet AS number);

46) A BGP connection is established between routers supporting 4-octet AS number, with one router using 2-octet AS number and the other router using 4-octet AS number.

Network Topology

As shown in the figure below, Router A, Router B and Router C are edge routers of three autonomous systems, and BGP connections have been established between them. Router A only supports 2-octet AS numbers; Router B and Router C support 4-octet AS numbers. The autonomous system of Router A uses a 2-octet AS number of 64496; the autonomous system of Router B uses a 2-octet AS number of 64497; the autonomous system of Router C uses a 4-octet number of 1.2.

Figure 18 BGP 4-Octet AS configuration



Configuration Tips

47) Router A cannot recognize the 4-octet AS number of 1.2 used by the autonomous system of Router C. When a neighbor is created, the reserved AS number of 23456 must replace 1.2 during the configuration of remote-as.

48) Although Router B supports 4-octet AS numbers, it still uses a 2-octet AS number. Therefore, the AS number of the peer can be used as remote-as while neighbor interconnection is created between Router A and Router B.

49) Router B can recognize the 4-octet AS number used by the autonomous system of Router C. The AS number of the peer can be used as remote-as while a neighbor is created.

Configuration Steps

1. Router A

```
Ruijie# conf t
Ruijie(config)# router bgp 64496
Ruijie(config-router)# neighbor 172.18.1.2 remote-as 64497
Ruijie(config-router)# neighbor 172.18.2.3 remote-as 23456
```

2. Router B

```
Ruijie# conf t
Ruijie(config)# router bgp 64497
Ruijie(config-router)# neighbor 172.18.1.1 remote-as 64496
Ruijie(config-router)# neighbor 172.18.3.3 remote-as 1.2
```

# Use "bgp asnotation dot" command to change the display format of 4-octet AS number

```
Ruijie(config-router)# bgp asnotation dot
Ruijie(config-router)# end
Ruijie# clear ip bgp *
```

3. Router C

```
Ruijie# conft
Ruijie(config)# router bgp 1.2
Ruijie(config-router)# neighbor 172.18.2.1 remote-as 64496
Ruijie(config-router)# neighbor 172.18.3.2 remote-as 64497
```

Verification

50) Display the state of neighbor connection on Router A:

```
Ruijie# show ip bgp summary
BGP router identifier 172.18.1.1, local AS number 64496
BGP table version is 1, main routing table version 1
Neighbor       V         AS MsgRcvd MsgSent  TblVer  InQ OutQ Up/Down  Statd
172.18.1.2     4     64497      7      7      1    0    0 00:03:04     0
172.18.2.3     4     23456      4      4      1    0    0 00:00:15     0
```

51) Display the state of neighbor connection on Router B:

```
Ruijie# show ip bgp summary
```

```
BGP router identifier 172.18.3.2, local AS number 64497
BGP table version is 1, main routing table version 1
Neighbor       V          AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  Statd
172.18.1.1     4      64496      7      7       1    0    0 00:03:04     0
172.18.3.2     4      65538      4      4       1    0    0 00:01:18     0
```

After executing "bgp notation dot" command, the following information will be displayed:

```
Ruijie# show ip bgp summary
BGP router identifier 172.18.3.2, local AS number 64497
BGP table version is 1, main routing table version 1
Neighbor       V          AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  Statd
172.18.1.1     4      64496      7      7       1    0    0 00:00:04     0
172.18.3.2     4        1.2      4      4       1    0    0 00:00:16     0
```

52) Display the state of neighbor connection on Router C:

```
Ruijie# show ip bgp summary
BGP router identifier 172.18.3.3, local AS number 65538
BGP table version is 1, main routing table version 1
Neighbor       V          AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  Statd
172.18.2.1     4      64496      7      7       1    0    0 00:00:15     0
172.18.3.2     4      65597      4      4       1    0    0 00:01:19     0
```

Configuring the BGP MDT Address Family

About the MDT Address Family

When using PIM-SSM to create Default-MDT during multicast VPN network configuration, configure a BGP MDT address family. Through routing based on the MDT address family, PE can discover other PE addresses and initiate the grating of SPT to other PEs (Configuration steps are detailed in "MD-SCG.doc").

Default Configurations

No address family is configured.

Configuring MDT address family

Configuring VRF Instance and Route-related Attributes

Before configuring the MDT address family, VRF instance and route-related attributes must be configured:

Command	Function
Ruijie # **configure terminal**	Enters global configuration mode
Ruijie(config)# **ip vrf** *VRF*	Creates a VRF named VRF1 and enters VRF mode.
Ruijie(config-vrf)# **rd** *rd-value*	Configures VRF RD value, which is identified using XX:XX format, such as RD 1:100. 1 refers to the AS ID of backbone network, while 100 is a numerical value specified by the user.
Ruijie(config-vrf)# **route-target** {**both** | **export** | **import**} *rt-value*	Configures route export and import RT attribute of VRF.
Ruijie(config-vrf)# {**export** | **import**} **map** *map*	Configures the route map for import and export routes, allowing policy-based filtering of import and export routes.
Ruijie(config-vrf)# **mdt default** *group-address*	Configures MDT group address of VRF.
Ruijie(config-vrf)# **exit**	Exit VRF mode and enter global configuration mode
Ruijie(config)# **interface** *IFNAME*	Enters interface configuration mode
Ruijie(config-if)# **ip vrf forwarding** *VRF*	Associates interface with VRF instance
Ruijie(config-if)# **ip address** *ip-address mask*	Configures an IP address for the interface
Ruijie(config-if)# **end**	Returns to privileged EXEC mode
Ruijie # **show running-config**	Verifies the configurations.
Ruijie # **write**	(Optional) Saves configurations.

Configuring MDT Address Family

The following part describes the steps of configuring an MDT address family:

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **router bgp** *asn-num*	Creates BGP and enters BGP configuration mode.
Ruijie(config-router)# **neighbor** *ip-address* **remote-as** *asn-number*	Configures BGP session.
Ruiije(config-router)# **neighbor** *ip-address* **update-source** *interface-name*	Configures the router to use interface address as the source address when MP-IBGP session is established. Usually, a Loopback interface address is used as the source address.
Ruiije(config-router)# **address**-**family ipv4** mdt	Enters MDT address family.
Ruijie(config-router-af)# **neighbor** *ip-address* **activate**	Activates the route to exchange MDT address family on BGP session.
Ruijie(config-router-af)# **neighbor** *ip-address* **next-hop-self**	Changes the next-hop route. This command can be executed on ASBR in OptionB.

Displaying Configurations

The BGP MDT address family can be viewed by executing the **show bgp ipv4 mdt** commands, as listed in the following table:

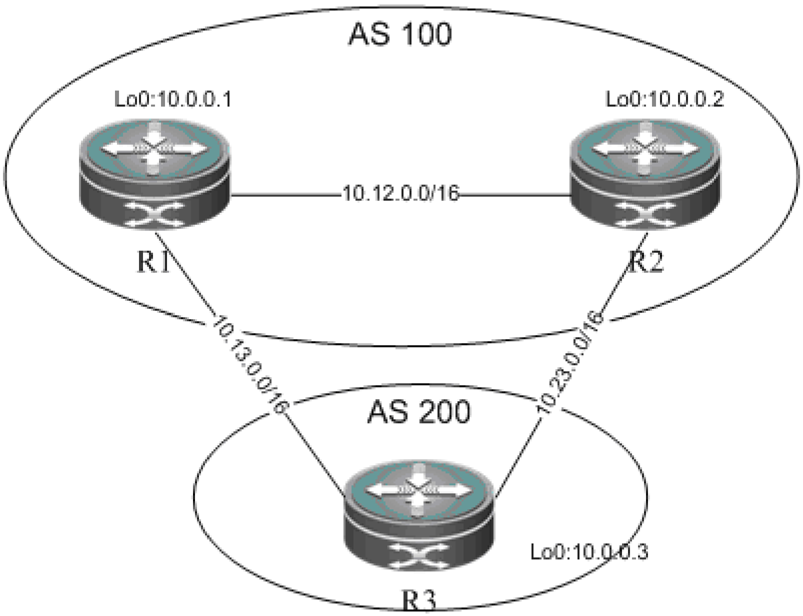
Command	Function
Ruijie # **show bgp ipv4 mdt all** [*ip-address* | neighbor [*ip-address*] | **summary**]	Displays all routes under all RDs, a specified route, neighbor information and summary information of the MDT address family.
Ruijie # **show bgp ipv4 mdt rd** *rd* [*ip-address*]	Displays all routes or a specified route under a specified RD of the MDT address family.

Typical Configuration Examples

Networking Requirements

R1 and R2 belong to the same AS. R3 belongs to another AS. Multicast VPN must be established between them, and BGP is used to transmit information about the MDT address family.

Network Topology



R1 and R2 belong to AS100. An IBGP connection is established between R1 and R2 to transmit routes of the MDT address family. R3 belongs to AS200 and establishes EBGP connections with R1 and R2 to transmit the MDT address family.

Configuration Tips

53) Configure VRF instances and associate with interfaces for network isolation;

54) Configure BGP routing protocol to advertise routes of the MDT address family

Configuration Steps

55) Configure VRF instances and associate with interfaces;

- R1

# Create a VRF instance named "VRF1"

```
Ruijie# config terminal
Ruijie(config)# ip vrf VRF1
Ruijie(config-vrf)# rd 100:1
Ruijie(config-vrf)# route-target both 123:123
Ruijie(config-vrf)# mdt default 232.1.1.1
Ruijie(config-vrf)# exit
```

# Associate Gi0/1 with VRF1

```
Ruijie(config)# interface gigabitEthernet 0/1
Ruijie(config-GigabitEthernet 0/1)# ip vrf forwarding VRF1
Ruijie(config-GigabitEthernet 0/1)# ip address 10.1.1.1 255.255.255.0
Ruijie(config-GigabitEthernet 0/1)# exit
```

```
R2 and R3 are configured to be the same as R1.
```

56) Configure the BGP routing protocol to advertise routes of the MDT address family

- R1

# Configure the MDT address family

```
Ruijie# configure terminal
Ruijie(config)# router bgp 100
```

# Configure R2 and R3 as BGP neighbors

```
Ruijie(config-router)# neighbor 10.0.0.2 remote-as 100
Ruijie(config-router)# neighbor 10.0.0.2 update-source loopback 0
Ruijie(config-router)# neighbor 10.13.0.3 remote-as 200
```

# Activate R2 and R3 under the MDT address family

```
Ruijie(config-router)# address-family ipv4 mdt
Ruijie(config-router-af)# neighobr 10.0.0.2 activate
Ruijie(config-router-af)# neighobr 10.13.0.3 activate
```

# Activate R2 and R3 under the VPNv4 address family

```
Ruijie(config-router)# address-family vpnv4
Ruijie(config-router-af)# neighobr 10.0.0.2 activate
Ruijie(config-router-af)# neighobr 10.13.0.3 activate
```

# Bind VRF to BGP

```
Ruijie(config-router)# address-family ipv4 vrf VRFi
Ruijie(config-router)# exit
R2 and R3 are configured to be the same as R1.
```

Verification

Take the following steps to verify configurations:

57) Verify the state of interfaces bound to VRF. Execute the **show ip vrf interface** to verify interface binding information and state.

```
Ruijie#show ip vrf interfaces
Interface           IP-Address     VRF                Protocol
GigabitEthernet 0/1  10.1.1.1       VRF1               up
```

58) Ensure MDT routes exist in BGP protocol, as shown below:

```
Ruijie#show bgp ipv4 mdt all
BGP table version is 1, local router ID is 10.0.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network         Next Hop           Metric  LocPrf     Weight Path
Route Distinguisher: 100:1
```

```
*> 10.0.0.1/32   0.0.0.0             0       32768      ?
*>i10.0.0.2/32   10.0.0.2            0       100        ?
*> 10.0.0.3/32   10.13.0.3           0                200 ?
Total number of prefixes 3
```

Configuring BGP Multi-Path Load Balancing

Understanding BGP Multi-Path Load Balancing

Overview

Multi-path load balancing means data packets are equally forwarded from a number of paths to the same network, and multiple next hops are present in the routing table. Based on the type of equivalent route, BGP multi-path load balancing comes under the following two categories:

- EBGP load balancing: through routes from EBGP neighbors.
- IBGP load balancing: through routes from IBGP neighbors.

Caution The protocol currently does not support load balancing between IBGP and EBGP routes.

Currently, IPv4 and IPv6 protocol stacks support multi-path load balancing, a maximum number of 32 equivalent next hops. The BGP does not limit the number of equivalent routes. This also applies to IBGP and EBGP load balancing.

Working Principle

BGP selects the route with the highest priority from multiple routes that are contained in the BGP routing table and destined to the same network. If several routes have the same priority and are all optimal, BGP will select the only one based on comparison and advertises the route to the forwarding plane for data flow control. When multi-path load balancing is enabled, BGP will list the routes with the same priority as the only optimal route as equivalent routes and advertise the optimal route and its equivalent routes to the forwarding plane for load balancing. Equivalent routes have the same priority and basic attributes.

Protocol Specifications

N/A

Default Configuration

BGP load balancing is disabled.

Configuring BGP Multi-Path Load Balancing

Configuring EBGP Load Balancing

Command	Function
Ruijie # **configure terminal**	Enters global configuration mode.

Ruijie(config)# **router bgp** *as-number*	Enables BGP and specifies the *as-number* range (1 to 4294967295).
Ruijie(config-router)# **maximum-paths ebgp** *number*	Sets the number of equivalent routes that support EBGP multi-path load balancing. *number* ranges from 1 to 32.
Ruijie(config-router)# **end**	Returns to privilege mode.
Ruijie # **write**	(Optional) Saves configuration.

Configuring IBGP Load Balancing

Command	Function
Ruijie # **configure terminal**	Enters global configuration mode.
Ruijie(config)# **router bgp** *as-number*	Enables BGP and specifies the *as-number* range (1 to 4294967295).
Ruijie(config-router)# **maximum-paths ibgp** *number*	Sets the number of equivalent routes that support IBGP multi-path load balancing. *number* ranges from 1 to 32.
Ruijie(config-router)# **end**	Returns to privilege mode.
Ruijie # **write**	(Optional) Saves configuration.

Configuring AS-PATH Loose Comparison

By default, equivalent routes must have equal AS-PATH. This requirement is too strict in some cases. For load balancing, AS-PATH loose comparison is recommended. Under the AS-APTH loose comparison mode, equivalent routes only need to have equal AS-PATH and AS-PATH in addition to meeting other criteria.

Command	Function
Ruijie # **configure terminal**	Enters global configuration mode.
Ruijie(config)# **router bgp** *number*	Enables BGP and specifies the *as-number* range (1 to 4294967295).
Ruijie(config-router)# **bgp bestpath as-path multipath-relax**	Enables BGP AS-PATH loose comparison.
Ruijie(config-router)# **maximum-paths ibgp** *number*	Sets the number of equivalent routes that support IBGP multi-path load balancing. *number* ranges from 1 to 32.
Ruijie(config-router)# **end**	Returns to privilege mode.
Ruijie # write	(Optional) Saves configuration.

Checking Configuration

Use the **show** command to view information about equivalent routes.

Command	Function
Ruijie # **show ip bgp**	Displays BGP routing information.
Ruijie # **show ip route**	Checks information in the core routing table.

Typical Configuration Examples of BGP Multi-Path Load Balancing

Configuring IBGP Non-Equivalent Load Balancing

Networking Requirements

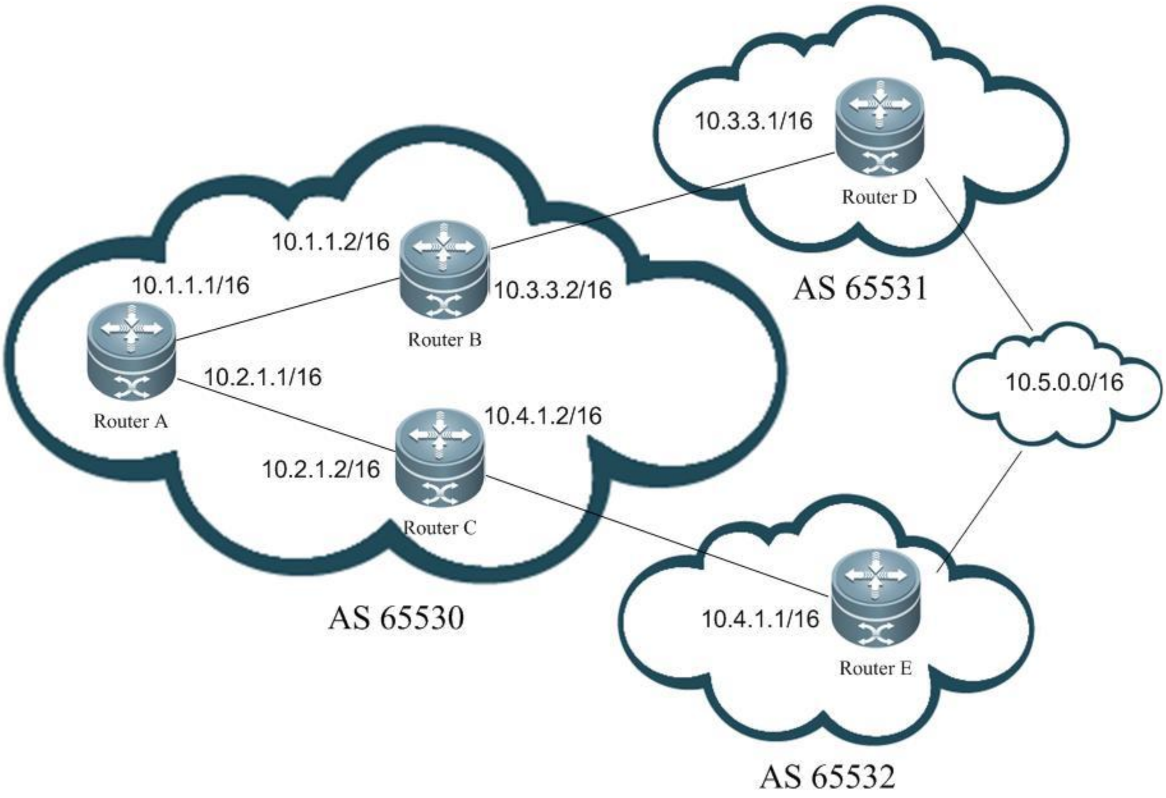
59) Achieves load balancing based on routes learned from IBGP neighbors;

60) Supports BGP AS-PATH loose comparison.

Network Topology

As shown in the following figure, Routers A, B and C belong to the same AS numbered 65530. Routers D and E belong to AS 65531 and 65532, which are linked by a BGP connection. AS 65531 and AS 65532 contain the route 10.5.0.0/16 with the same prefix and send the route to AS 65530. Router A learn the route 10.5.0.0/16 from Router B and Router C.

Figure 19 Configuring BGP ECMP



Configuration Precautions

61) Enables IBGP load balancing on Router A, and AS-PATH loose comparison.

62) Routers B and D, Routers C and E are connected to EBGP neighbors through a single hop.

63) Routers B and C consider Router A as IBGP neighbors.

Configuration Steps

64) Configuration on Router A

```
Ruijie# conf t
Ruijie(config)# interface fastEthernet 0/0
Ruijie(config-if-FastEthernet 0/0)# ip address 10.1.1.1 255.255.0.0
Ruijie(config-if-FastEthernet 0/0)# exit
Ruijie(config)# interface fastEthernet 0/1
Ruijie(config-if-FastEthernet 0/1)# ip address 10.2.1.1 255.255.0.0
Ruijie(config-if-FastEthernet 0/1)# exit
Ruijie(config)# ip route 10.3.0.0 255.255.0.0 10.1.1.2
Ruijie(config)# ip route 10.4.0.0 255.255.0.0 10.2.1.2
Ruijie(config)# router bgp 65530
Ruijie(config-router)# neighbor 10.1.1.2 remote-as 65530
Ruijie(config-router)# neighbor 10.2.1.2 remote-as 65530
Ruijie(config-router)# bgp maximum-paths ibgp 2
Ruijie(config-router)# bgp bestpath as-path multipath-relax
```

65) Configuration on Router B

```
Ruijie# conf t
Ruijie(config)# interface fastEthernet 0/0
Ruijie(config-if-FastEthernet 0/0)# ip address 10.1.1.2 255.255.0.0
Ruijie(config-if-FastEthernet 0/0)# exit
Ruijie(config)# interface fastEthernet 0/1
Ruijie(config-if-FastEthernet 0/1)# ip address 10.3.1.2 255.255.0.0
Ruijie(config-if-FastEthernet 0/1)# exit
Ruijie(config)# router bgp 65530
Ruijie(config-router)# neighbor 10.1.1.1 remote-as 65530
Ruijie(config-router)# neighbor 10.3.1.1 remote-as 65531
```

66) Configuration on Router C

```
Ruijie# conf t
Ruijie(config)# interface fastEthernet 0/0
Ruijie(config-if-FastEthernet 0/0)# ip address 10.2.1.2 255.255.0.0
Ruijie(config-if-FastEthernet 0/0)# exit
Ruijie(config)# interface fastEthernet 0/1
Ruijie(config-if-FastEthernet 0/1)# ip address 10.4.1.2 255.255.0.0
Ruijie(config-if-FastEthernet 0/1)# exit
Ruijie(config)# router bgp 65530
Ruijie(config-router)# neighbor 10.2.1.1 remote-as 65530
Ruijie(config-router)# neighbor 10.4.1.1 remote-as 65532
```

67) Configuration on Router D

```
Ruijie# conf t
Ruijie(config)# interface fastEthernet 0/0
Ruijie(config-if-FastEthernet 0/0)# ip address 10.3.1.1 255.255.0.0
Ruijie(config-if-FastEthernet 0/0)# exit
Ruijie(config)# interface loopback 1
Ruijie(config-if)#ip address 10.5.1.1 255.255.0.0
Ruijie(config-if-FastEthernet 0/1)# exit
Ruijie(config)# router bgp 65531
Ruijie(config-router)# neighbor 10.3.1.2 remote-as 65530
Ruijie(config-router)# redistribute connected
```

68) Configuration on Router E

```
Ruijie# conf t
Ruijie(config)# interface fastEthernet 0/0
Ruijie(config-if-FastEthernet 0/0)# ip address 10.4.1.1 255.255.0.0
Ruijie(config-if-FastEthernet 0/0)# exit
Ruijie(config)# interface loopback 1
Ruijie(config-if)#ip address 10.5.1.2 255.255.0.0
Ruijie(config-if-FastEthernet 0/1)# exit
Ruijie(config)# router bgp 65532
Ruijie(config-router)# neighbor 10.4.1.2 remote-as 65530
Ruijie(config-router)# redistribute connected
```

Checking Configuration

69) Check the status of neighbor connection on Router A:

```
Ruijie#show ip bgp summary
BGP router identifier 10.2.1.1, local AS number 65530
BGP table version is 9
2 BGP AS-PATH entries
0 BGP Community entries
3 BGP Prefix entries (Maximum-prefix:4294967295)
Neighbor       V   AS      MsgRcvd   MsgSent   TblVer   InQ   OutQ   Up/Down    State/PfxRcd
172.16.23.140   4   65530   29        25        8       0     0      00:18:48   2
172.16.23.141   4   65530   24        21        8       0     0      00:17:58   2
Total number of neighbors 2
```

70) Check BGP routes on Router A.

```
Ruijie#show ip bgp
BGP table version is 9, local router ID is 10.2.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
            S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network        Next Hop   Metric   LocPrf   Weight   Path
*>i10.3.0.0/16    10.3.1.1    0       100      0       65531 ?
*>i10.4.0.0/16    10.4.1.1    0       100      0       65532 ?
* i10.5.0.0/16    10.3.1.1    0       100      0       65531 ?
*>i              10.4.1.1   0       100      0       65532 ?
Total number of prefixes 3
```

71) Check BGP route 10.5.0.0 on Router A:

```
Ruijie#show ip bgp 10.5.0.0
BGP routing table entry for 10.5.0.0/16
Paths: (2 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  65532
    10.4.1.1 from 10.2.1.2 (172.16.24.1)
      Origin incomplete, metric 0, localpref 100, valid, internal, multipath, best
      Last update: Mon Mar 21 03:45:14 2011
  65531
    10.3.1.1 from 10.1.1.2 (172.16.25.1)
      Origin incomplete, metric 0, localpref 100, valid, internal, multipath
      Last update: Mon Mar 21 03:45:14 2011
```

72) Check routes in Router A’s core routing table:

```
Ruijie#show ip route
```

```
Codes:  C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default
Gateway of last resort is no set
C    10.1.0.0/16 is directly connected, FastEthernet 0/0
C    10.1.1.1/32 is local host.
C    10.2.0.0/16 is directly connected, FastEthernet 0/1
C    10.2.1.1/32 is local host.
S    10.3.0.0/16 [1/0] via 10.1.1.2
S    10.4.0.0/16 [1/0] via 10.2.1.2
B    10.5.0.0/16 [200/0] via 10.3.1.1, 00:27:56
              [200/0] via 10.4.1.1, 00:27:56
```

Configuring BGP/MPLS VPN

Please refer to the section "BGP/MPLS L3VPN Configuration" in the "MPLS Configuration Guideline" for details.

Configuring BGP/MVPN

Please refer to the section "Multicast VPN Configuration" in the "Multicast VPN Configuration Guideline" for details.

Configuring BGP Fast-Reroute

BGP Fast-Reroute Overview

IP Fast-Reroute Overview

With high-speed development of IP technologies and application of various complex services, the requirements for network security and stability become increasingly higher. Especially, certain real-time services (audios and videos) are sensitive to network running status and may be largely affected by unstable networks. Therefore, more and more focus and importance are attached to network reliability. With these requirements, the IP Fast-Reroute function comes into being. It is intended to use a backup link to maintain data forwarding during route platform convergence after a faulty link is detected, in order to achieve the ideal targets of "zero delay" and "zero loss" in packet forwarding.

Working Principle of BGP Fast-Reroute

According to the description in RFC 4271, BGP uses a routing policy to select a best route for forwarding local data and advertises the route to neighbors. After the BGP Fast-Reroute function is used, BGP selects a backup route for each best route. After BFD Fast-Reroute detects that the master link is faulty, it switches the data to the originally calculated backup link for forwarding. After route convergence is completed, data is switched to the best route re-calculated for forwarding. In this way, BGP Fast-Reroute can avoid route disconnection due to a link fault before BGP route convergence is completed.

Default Configuration

BGP Fast-Reroute is disabled by default.

Configuring BGP Fast-Reroute

Enable BGP Fast-Reroute by using the following steps:

	Command	Description
Step 1	Ruijie# **configure terminal**	Enters global configuration mode.
Step 2	Ruijie(config)# **router bgp** *as-number*	Enters BGP routing configuration mode.
Step 3	Ruijie(config-router)# **address-family ipv4 unicast**	Enters the BGP IPv4 address family.
Step 4	Ruijie(config-router-af)# **bgp fast-reroute**	Enables BGP Fast-Reroute.
Step 5	Ruijie(config-router-af)#**exit**	Returns to BGP routing configuration mode.
Step 6	Ruijie(config-router)#**neighbor** *peer-address* **fall-over bfd**	Configures a BFD session with a BGP neighbor.
Step 7	Ruijie(config-router)#**exit**	Returns to global configuration mode.

	Command	Description
Step 8	Ruijie(config)#**bfd bind bgp peer-ip** *ip-address* **interface** *interface-type interface-index* **source-ip** *ip-address*	(Optional) Configure a BFD session when it is impossible to detect failure of the master link through a BFD session with a BGP neighbor. It is not recommended to use the two commands for configuring the BFD session at the same time.
Step 9	Ruijie(config)# **end**	Returns to privileged EXEC mode.
Step 10	Ruijie # **show running-config**	Verifies the configuration.
Step 11	Ruijie # **write**	(Optional) Saves the configuration.

Product Compatibility

All products that support BGP and IP Fast-Reroute support the above configuration commands.

Caution

BGP Fast-Reroute is subject to the following constraints:

1) BGP Fast-Reroute is supported only in the IPv4 Unicast and IPv4 VRF address families of BGP.

2) Only one backup route can be generated and the next hop of the backup route cannot be the same as that of the preferred route.

3) A backup next hop cannot be generated for an ECMP route.

4) In BGP IPv4 VRF configuration mode, BGP Fast-Reroute has a lower priority than VPN Fast-Reroute. That is, if VPN Fast-Reroute is enabled in VRF mode, BGP Fast-Reroute takes effect only when VPN Fast-Reroute fails to calculate a backup route.

Configuration Example

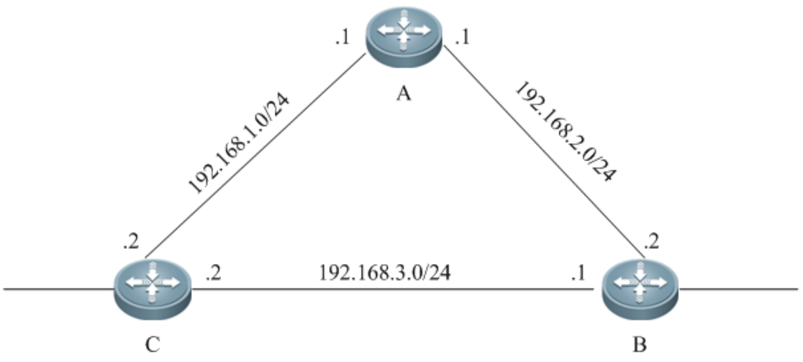
Configuring EBGP Fast-Reroute

Networking Requirements

As shown in the following figure, the three routers A, B and C belong to different autonomous areas (ASs). They are interconnected to each other through BGP. It is required that when the link between B and C is faulty, traffic can be fast switched to the link between C and A, and the traffic path is C→A→B.

Network Topology

Figure 0-1 EBGP Fast-Reroute



Configuration Steps

1) Configure router A.

```
Ruijie# conf t
Ruijie(config)# interface GigabitEthernet 0/1
Ruijie(config-if-GigabitEthernet 0/1)# ip address 192.168.1.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 0/1)# exit
Ruijie(config)# interface GigabitEthernet 0/2
Ruijie(config-if-GigabitEthernet 0/2)# ip address 192.168.2.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 0/2)# exit
Ruijie(config)# router bgp 100
Ruijie(config-router)# neighbor 192.168.1.2 remote-as 300
Ruijie(config-router)# neighbor 192.168.2.2 remote-as 200
Ruijie(config-router)# redistribute connect
```

2) Configure router B.

```
Ruijie# conf t
Ruijie(config)# interface GigabitEthernet 0/1
Ruijie(config-if-GigabitEthernet 0/1)# ip address 192.168.3.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 0/1)# bfd interval 200 min_rx 200 multiplier 5
Ruijie(config-if-GigabitEthernet 0/1)# exit
Ruijie(config)# interface GigabitEthernet 0/2
Ruijie(config-if-GigabitEthernet 0/2)# ip address 192.168.2.2 255.255.255.0
Ruijie(config-if-GigabitEthernet 0/2)# exit
Ruijie(config)# router bgp 200
Ruijie(config-router)# neighbor 192.168.3.2 remote-as 300
Ruijie(config-router)# neighbor 192.168.3.2 fall-over bfd
Ruijie(config-router)# neighbor 192.168.2.1 remote-as 100
Ruijie(config-router)# redistribute connect
```

3) Configure router C.

```
Ruijie# conf t
```

```
Ruijie(config)# interface GigabitEthernet 0/1
Ruijie(config-if-GigabitEthernet 0/1)# ip address 192.168.1.2 255.255.255.0
Ruijie(config-if-GigabitEthernet 0/1)# exit
Ruijie(config)# interface fastEthernet 0/2
Ruijie(config-if-GigabitEthernet 0/2)# ip address 192.168.3.2 255.255.0.0
Ruijie(config-if-GigabitEthernet 0/2)# bfd interval 200 min_rx 200 multiplier 5
Ruijie(config-if-GigabitEthernet 0/2)# exit
Ruijie(config)# router bgp 300
Ruijie(config-router)# neighbor 192.168.1.1 remote-as 100
Ruijie(config-router)# neighbor 192.168.3.1 remote-as 200
Ruijie(config-router)# neighbor 192.168.3.1 fall-over bfd
Ruijie(config-router)# address-family ipv4 unicast
Ruijie(config-router-af)# bgp fast-reroute
Ruijie(config-router-af)# redistribute connect
```

Verification

4) Display the neighbor connection status on router C.

```
Ruijie# show ip bgp summary
BGP router identifier 10.10.10.10, local AS number 300
BGP table version is 12
4 BGP AS-PATH entries
0 BGP Community entries
3 BGP Prefix entries (Maximum-prefix:4294967295)

Neighbor       V        AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
192.168.1.1     4        100     76     77       12    12    0 00:59:27       3
192.168.3.1     4        200     30     30       12    12    0 00:19:03       3

Total number of neighbors 2
```

5) Display the BGP route on router C.

```
Ruijie# show ip bgp
BGP table version is 12, local router ID is 10.10.10.10
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
            S Stale, b - backup entry
Origin codes: i - IGP, e - EGP, ? - incomplete

  Network        Next Hop          Metric     LocPrf     Weight Path
*  192.168.1.0     192.168.3.1           0                    0 200 ?
*               192.168.1.1           0                    0 100 ?
*>              0.0.0.0              0               32768    ?
*> 192.168.2.0     192.168.3.1           0                    0 200 ?
*b              192.168.1.1           0                    0 100 ?
```

```
*  192.168.3.0    192.168.3.1            0                   0 200 ?
*                192.168.1.1            0                   0 100 200 ?
*>               0.0.0.0               0                32768    ?

Total number of prefixes 3
```

6) Display the BGP route 192.168.2.0 on router C.

```
Ruijie# show ip bgp 192.168.2.0
BGP routing table entry for 192.168.2.0/24
Paths: (2 available, best #1, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
  192.168.1.1
  200
    192.168.3.1 from 192.168.3.1 (3.3.3.3)
      Origin incomplete, metric 0, localpref 100, valid, external, best
      Last update: Tue Oct  5 00:26:52 1971

  100
    192.168.1.1 from 192.168.1.1 (44.44.44.44)
      Origin incomplete, metric 0, localpref 100, valid, external, backup
      Last update: Mon Oct  4 23:46:28 1971
```

7) Display the route in the core routing table on router C.

```
Ruijie# show ip route

Codes:  C - connected, S - static, R - RIP, B - BGP
        O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
C    192.168.1.0/24 is directly connected, GigabitEthernet 1/9
C    192.168.1.2/32 is local host.
B    192.168.2.0/24 [20/0] via 192.168.3.1, 00:21:39
C    192.168.3.0/24 is directly connected, GigabitEthernet 1/11
C    192.168.3.2/32 is local host.
```

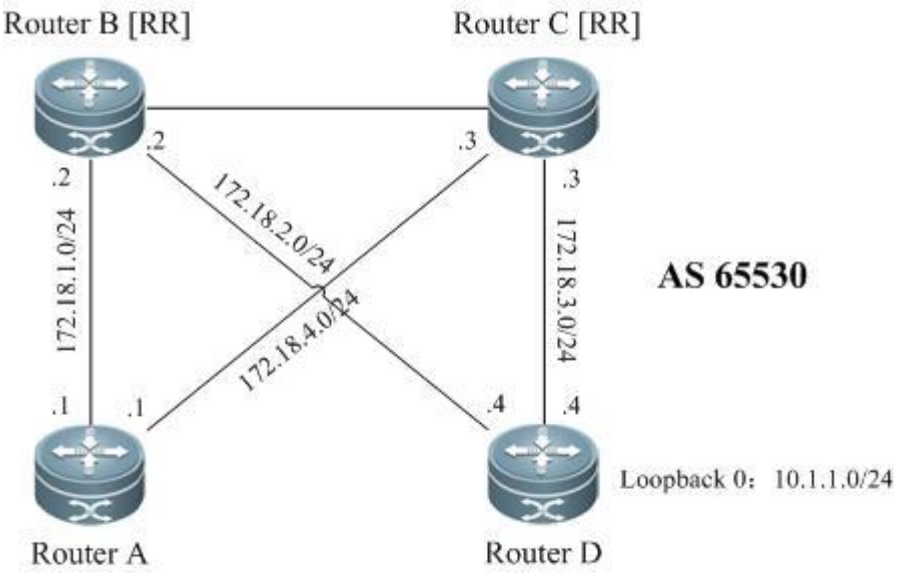
Configuring IBGP Fast-Reroute

Networking Requirements

As shown in the following figure, the router routers A, B, C and D belong to the same AS. B and C serve as the route reflectors and establish IBGP neighbors with A and D respectively. The route 10.1.1.0/24 advertised by D will be reflected to A through B and C at the same time. Normally, A prefers the link A->B->D to reach the network segment 10.1.1.0/24. It is required that when the link A->B->D is faulty, traffic can be fast switched to the link A->C->D.

Network Topology

Figure 0-2 IBGP Fast-Reroute



Configuration Steps

8) Configure router A.

```
Ruijie# conf t
Ruijie(config)# interface GigabitEthernet 0/1
Ruijie(config-if-GigabitEthernet 0/1)# ip address 172.18.1.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 0/1)# bfd interval 200 min_rx 200 multiplier 5
Ruijie(config-if-GigabitEthernet 0/1)# exit
Ruijie(config)# interface GigabitEthernet 0/2
Ruijie(config-if-GigabitEthernet 0/2)# ip address 172.18.4.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 0/2)# exit
Ruijie(config)# router bgp 65530
Ruijie(config-router)# neighbor 172.18.1.2 remote-as 65530
Ruijie(config-router)# neighbor 172.18.4.3 remote-as 65530
Ruijie(config-router)# bgp fast-reroute
Ruijie(config-router)# exit
Ruijie(config)# bfd bind bgp peer-ip 172.18.2.4 interface GigabitEthernet 0/1 source-ip
172.18.1.1
```

9) Configure router B.

```
Ruijie# conf t
Ruijie(config)# interface GigabitEthernet 0/1
Ruijie(config-if-GigabitEthernet 0/1)# ip address 172.18.1.2 255.255.255.0
Ruijie(config-if-GigabitEthernet 0/1)# exit
Ruijie(config)# interface GigabitEthernet 0/2
Ruijie(config-if-GigabitEthernet 0/2)# ip address 172.18.2.2 255.255.255.0
Ruijie(config-if-GigabitEthernet 0/2)# exit
Ruijie(config)# router bgp 65530
Ruijie(config-router)# neighbor 172.18.1.1 remote-as 65530
Ruijie(config-router)# neighbor 172.18.2.4 remote-as 65530
Ruijie(config-router)# address-family ipv4 unicast
Ruijie(config-router-af)# neighbor 172.18.1.1 route-reflector-client
Ruijie(config-router-af)# neighbor 172.18.2.4 route-reflector-client
Ruijie(config-router-af)# end
```

10) Configure router C.

```
Ruijie# conf t
Ruijie(config)# interface GigabitEthernet 0/1
Ruijie(config-if-GigabitEthernet 0/1)# ip address 172.18.4.3 255.255.255.0
Ruijie(config-if-GigabitEthernet 0/1)# exit
Ruijie(config)# interface GigabitEthernet 0/2
Ruijie(config-if-GigabitEthernet 0/2)# ip address 172.18.3.3 255.255.255.0
Ruijie(config-if-GigabitEthernet 0/2)# exit
Ruijie(config)# router bgp 65530
Ruijie(config-router)# neighbor 172.18.4.1 remote-as 65530
Ruijie(config-router)# neighbor 172.18.3.4 remote-as 65530
Ruijie(config-router)# address-family ipv4 unicast
Ruijie(config-router-af)# neighbor 172.18.4.1 route-reflector-client
Ruijie(config-router-af)# neighbor 172.18.3.4 route-reflector-client
Ruijie(config-router-af)# end
```

11) Configure router D.

```
Ruijie# conf t
Ruijie(config)# interface GigabitEthernet 0/1
Ruijie(config-if-GigabitEthernet 0/1)# ip address 172.18.2.4 255.255.255.0
Ruijie(config-if-GigabitEthernet 0/1)# bfd interval 200 min_rx 200 multiplier 5
Ruijie(config-if-GigabitEthernet 0/1)# exit
Ruijie(config)# interface GigabitEthernet 0/2
Ruijie(config-if-GigabitEthernet 0/2)# ip address 172.18.3.4 255.255.255.0
Ruijie(config-if-GigabitEthernet 0/2)# exit
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 10.1.1.1 255.255.255.0
Ruijie(config-if-Loopback 0)# exit
Ruijie(config)# router bgp 65530
Ruijie(config-router)# neighbor 172.18.2.2 remote-as 65530
Ruijie(config-router)# neighbor 172.18.3.3 remote-as 65530
Ruijie(config-router)# network 10.1.1.0 mask 255.255.255.0
Ruijie(config-router)# exit
Ruijie(config)# bfd bind bgp peer-ip 172.18.1.1 interface GigabitEthernet 0/1 source-ip
172.18.2.4
```

Verification

12) Display the neighbor connection status on router A.

```
Ruijie# show ip bgp summary
BGP router identifier 10.1.1.2, local AS number 300
BGP table version is 12
0 BGP AS-PATH entries
0 BGP Community entries
1 BGP Prefix entries (Maximum-prefix:4294967295)

Neighbor       V        AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
172.18.1.2     4      65530     76     77       12    12    0 00:59:27       1
172.18.4.3     4      65530     30     30       12    12    0 00:19:03       1

Total number of neighbors 2
```

13) Display the BGP route on router A.

```
Ruijie# show ip bgp
BGP table version is 12, local router ID is 10.1.1.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
            S Stale, b - backup entry
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network         Next Hop          Metric     LocPrf     Weight Path
*>i10.1.1.0         172.18.2.4            0        100         0    i
*bi              172.18.3.4            0        100         0    i

Total number of prefixes 3
```

14) Display the BGP route 10.1.1.0 on router A.

```
Ruijie# show ip bgp 10.1.1.0
BGP routing table entry for 10.1.1.0/24
Paths: (2 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  Local
    172.18.2.4 (metric 10) from 172.18.1.2 (3.3.3.3)
      Origin incomplete, metric 0, localpref 100, valid, external, best
```

```
     Last update: Tue Oct  5 00:26:52 1971

  Local
   172.18.3.4 (metric 20) from 172.18.4.3 (44.44.44.44)
     Origin incomplete, metric 0, localpref 100, valid, external, backup
     Last update: Mon Oct  4 23:46:28 1971
```

15) Display the route in the core routing table on router A.

```
Ruijie# show ip route

Codes:  C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
C   172.18.1.0/24 is directly connected, GigabitEthernet 1/1
C   172.18.1.1/32 is local host.
C   172.18.4.0/24 is directly connected, GigabitEthernet 1/2
C   172.18.4.1/32 is local host.
B   10.1.1.0/24 [200/0] via 172.18.2.4, 00:21:39
```

Configuring IS-IS

Understanding IS-IS Protocol

Overview

IS-IS (Intermediate System-to-Intermediate System) is a routing protocol defined in ISO10589. It was initially a dynamic routing protocol designed by ISO for CLNP (Connectionless Network Protocol). With IP getting more and more popular, IETF enables IS-IS to support IP in RFC1195 and develops IS-IS into Integrated IS-IS. After years of development, Integrated IS-IS has become a scalable, robust and easy-to-use IGP protocol, which is applicable to IP and ISO CLNS based dual-environment network.

As a link-state protocol, IS-IS has certain features shared by link-state protocols. It discovers and maintains adjacencies by sending Hello packets, and advertises its own link state by sending LSP (Link State PDU) to its neighbors. IS-IS has a two-level hierarchy (level 1 and level 2 routing), with all devices at the same level having the same LSDB, which stores the LSP generated by all devices at the same level. In this way, all devices at the same level are aware of the network topology of their level, and each device uses Dijkstra SPF algorithm to optimize route calculation, select path and achieve fast convergence.

Hierarchical Structure of IS-IS Network

Figure 1

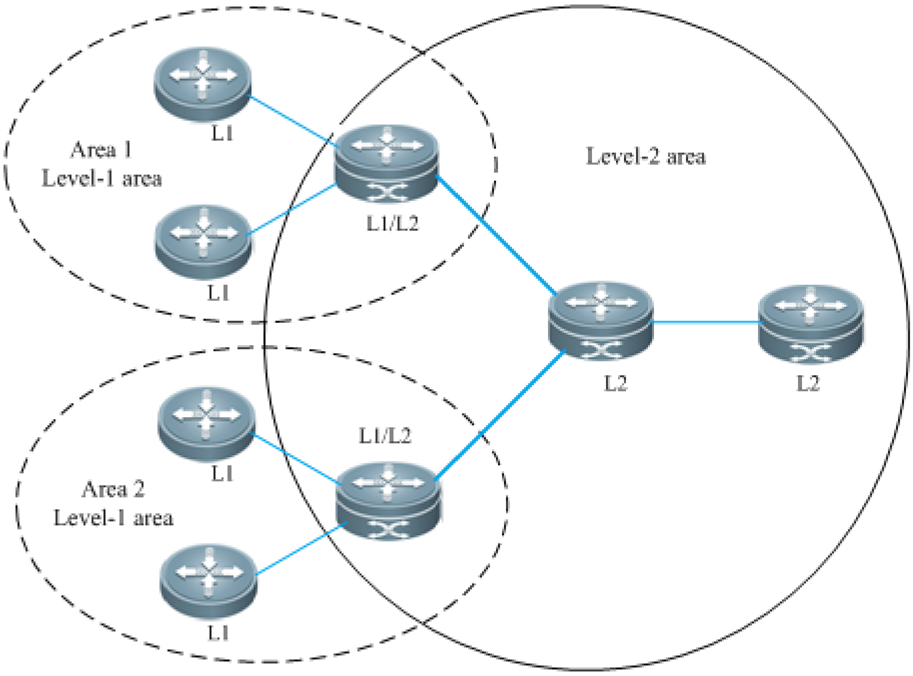


Figure 1 IS-IS hierarchy

This network is divided into Level-1 and Level-2. All nodes for exchanging information among devices in the same area form Level-1. All intra-area devices are aware of the network topology of entire area and carry out Inter-area data exchange. Level-1-2 devices are the boundary devices to connect different areas. Inter-area connection is achieved by connecting Level-2 devices, while the boundary devices of respective areas jointly form a backbone network (Level-2). Inter-area data exchange is carried out at level-2.

Level-1 devices only concern about the topology structure of the local area, including all nodes and next-hop devices reaching these nodes in the local area. Level-1 device accesses other areas through the Level-2 device, and forwards data packets in the destination network outside the area to the closest Level-2 device.

Address Encoding of IS-IS Protocol

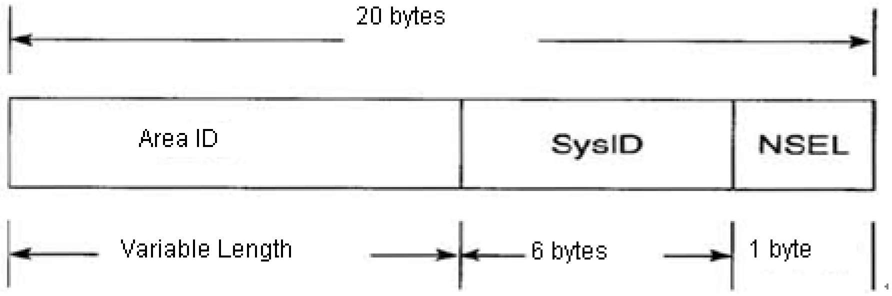


Figure 2 NET address format

IS-IS protocol address is called NET, which can be divided into three parts: Area address, System ID and NSAP selector. The total length of NSAP address ranges from 8 to 20 bytes.

The length reserved for area address is variable. The area ID is the length of route domain, and is fixed in the route domain. The length of area address ranges from 1 to 13 bytes.

The length of System ID is 6 bytes, and is unique in the autonomous system.

NSAP is the network selector, and is sometimes called SEL, with length being 1 byte. In IS-IS, SEL is usually set to 00 to represent the routing device.

IS-IS Packet Types

There are three types of packets:

- Link-state PDUs (LSP)
- IS-IS Hello PDUs

Sequence number PDUs (SNP)
Link-state PDUs (LSP) are used to advertise link-state logs within the area. They can be divided into: Level 1 Link State PDU and Level 2 Link State PDU. LSP will only be flooded at its own level.

IS-IS Hello PDUs (IIH PDU) are used to maintain adjacencies. Hello PDUs will send multicast MAC address to detect whether IS-IS is operated in other systems.

Sequence number PDUs (SNP) can be divided into CSNP and PSNP.

Complete sequence number PDU (CSNP) is used to synchronize LSDB. In a broadcast network, DIS will send CSNP packets once every 10 seconds by default. In a point-to-point network, CSNP packets will only be sent once after adjacency is formed.

Partial sequence number PDU (PSNP) is also used to synchronize LSDB.

DIS

DIS: Designated IS, the designated routing device in on the broadcast network, equivalent to the DR in OSPF.

Pseudonode: The pseudonode is generated by DIS and establishes contacts with all devices in the network.

DIS will model the multi-access link as a pseudonode to create pseudonode LSPs. All routing devices on the local network contact with the pseudonode, and no direct contact between them is allowed. The broadcast subnet and NBMA network are regarded as a pseudonode externally. All non-DIS devices on the network will report their link state to the DIS, which will report the link state on behalf of all ISs on the entire network. The reason to elect DIS is the same as the reason to elect DR in OSPF: to reduce unnecessary adjacencies and routing information exchange.

DIS is created through election. The DIS election in IS-IS is pre-emptive, which is different from DR in OSPF.

The result of DIS election can be controlled by configuring the "Priority" of interface. The one with the highest "Priority" value will be elected.

TLVs Supported by IS-IS

Currently, Ruijie IS-IS supports the following TLV codes:

TLV CODE	Description
Code=1	Area addresses
Code=2	Priority level information of IS neighbor
Code =3	ES neighbors
Code=6	MAC address of IS neighbor
Code=8	Padding
Code=9	LSP entries
Code=10	Authentication information
Code=14	Buffer size of source LSP
Code=22	Extended IS reachability
Code=128	IP internal reachability information
Code=129	Protocols supported
Code =130	IP external reachability information
Code=131	IDRP information
Code=132	IP interface address
Code=133	Authentication information

Code=135	Extended IP reachability TLV
Code=137	Dynamic host name
Code = 211	Graceful Restart
Code=232	IPV6 interface
Code =236	IPV6 IP Reachability TLV
Code =240	P2P 3-way handshake TLV

LSP Fragments Extension

IS-IS informs devices of link-state information by flooding LSP packets. The size of LSP packets is restricted by link MTU and cannot be extended. When the information to be informed exceeds the size of a LSP packet, IS-IS will create LSP fragments to carry new link-state information. According to the ISO standard, the LSP fragment is recognized by the 1-byte LSP Number. Therefore, the maximal number of LSP fragments produced by an IS-IS node is 256.

There are several reasons causing 256 fragments not enough:

- New TLV or Sub-TLV extended by new application, such as TE
- Constant expansion of network scale
- Informing routes with smaller chip or redistributing other routes to IS-IS.

When LSP fragments are filled, subsequent routing information and neighbor information will be discarded directly. There will be network anomaly, such as routing blackhole or routing loops. LSP fragments need to be extended to carry more link-state information to ensure normal operation of the network.

Definitions of fragments extension are listed as follows:

- Normal system-id: It refers to the current system ID defined by ISO, which is used to form adjacency and learn routes. “Nornal” differentiates this kind of system-id from additional system-id produced by fragments extension.
- Additional system-id: It is configured by the administrator and used to extend LAP, in comparison with normal system-id. Additional system-id does not appear in Hello packets for adjacency formation. Except that, additional system-id adopts the same rules as normal system-id, for example, it must be unique and cannot be repeated in the whole intra-area.
- Originating System: It refers to the routing device running the IS-IS protocol. It is in comparison with the virtual system identified by the additional system ID.
- Virtual System/Vritual IS: It refers to the system identified by additional system-id, which is used to generate extended LSP. RFC proposes this notion and differentiate it from the originating system. Every virtual system generates up to 256 LSP fragment packets. The administrator can configure several additional system IDs , which represent virtual systems, to generate more LSP fragment packets to meet the demand.
- Original LSP: The LSP packet is generated by the originating system. The system-id is normal system-id.
- Extended LSP: The LSP packet is generated by the virtual system. The system-id is additional system-id.

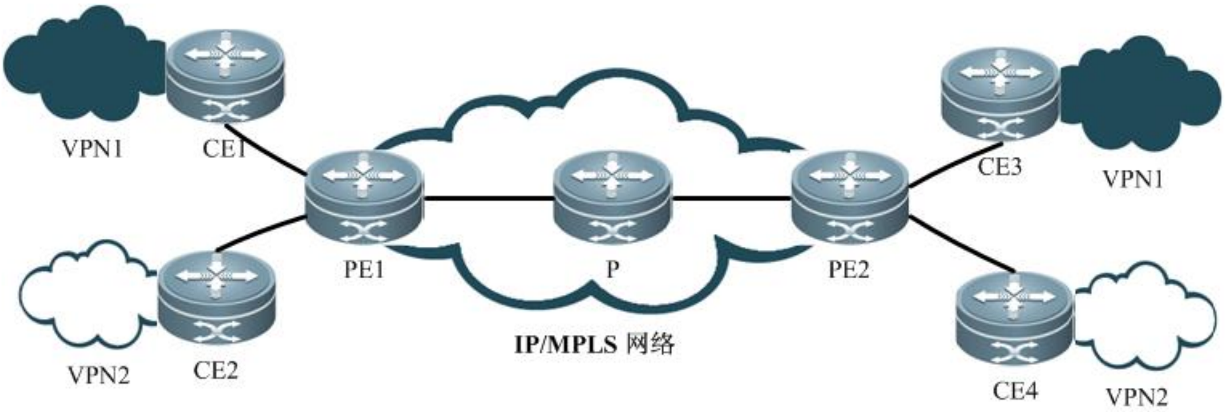
IS-IS can inform devices of more link-state information with extended LSP by setting additional system-id and enabling fragments extension. Every virtual system can be regarded as a virtual routing device which establishes adjacency with the originating system. The metric between them is 0. Extended LSP is the LSP packet released by the neighbor of originating system, namely the virtual system,

IS-IS VRF

VRF is short for VPN Routing and Forwading. It is mainly used to perform local routing, segregate data packets and address routing conflicts caused by VPNs using the same prefix. Most IPv4 and IPv6 VPNs are MPLS VPN. Combined with MPLS's advantage in service quality and security guarantee, MPLS VPN has become the preferred solution to enable interconnection among branches of enterprises and industries in different areas.

The following figure is a typical VRF networking application, which is to enable VPN segregation control by configuring VRF on PE devices.

Figure 1-3 Enabling VPN segregation control by configuring VRF on PE devices



As figure 1-3 shows, two site users (CE1 and CE3) under VPN1 should be able to visit each other. Two site users (CE2 and CE4) under VPN2 should be able to visit each other. VPN1 and VPN2 should not be able to visit each other for two reasons:

- The two VPNs belong to different users or departments. Mutual visit is prevented for security's sake.
- There may be the same IP address on VPN1 and VPN2.

CE is used to connect the user network to PE and exchange VPN routing information with PE: release local routes to PE and learn remote site routes from PE.

PE is used to learn routing from directly connected CE and exchange learned VPN routes with other PEs through BGP. The PE device is responsible for the access of VPN business.

The device P is a device that is not directly connected with CE on the operator network. The device is only required to support MPLS forwarding and cannot sense VPN.

The IS-IS routing protocol runs between PE and CE to enable VRF-based routes learning. PE and CE only learn routes within the same VPN to enable VPNs segregation control.

IS-IS Definitions

- ES: End System refers to non-router devices, such as host.
- IS: Intermediate System refers to router devices, the basic unit sending routing information and generating routes in the IS-IS protocol.
- ES-IS: End System-to-Intermediate System, an OSI protocol that defines how end systems (ES) and intermediate systems (IS) learn about each other.

- Domain: routing domain. In one routing domain, a group of ISs will exchange routing information through the same routing protocol.
- Area: a routing sub-domain. One routing domain can be divided into multiple areas.
- CSNP: Complete sequence number PDU, sent by DIS every 10 seconds on the broadcast network to synchronize link state.
- PSNP: Partial sequence number PDU, sent on the point-to-point link to acknowledge receipt of an LSP or on the broadcast network to request an LSP.
- ENPA: attached subnet point that provides subnet services.
- CLNP: Connectionless Network Protocol, an IP-alike OSI protocol to transmit data and error messages in the network layer.
- CNLS: Connectionless Network Service is the solution to unreliable connection which doesn't need the circuit to be established before data transmission.
- DIS: Designated Intermediate System, which is similar to the DR in OSPF. It floods LSPs to other devices on the LAN. Unlike OSPF, DIS forms adjacencies with other devices which also form adjacencies between each other.
- Hello: This packet is used to establish and maintain adjacencies.
- LSP: Link-state PDU, which is similar to the LSA in OSPF, but LSP doesn't rely on TCP/IP protocol information. There are different LSPs for different routes, such as L1 LSP and L2 LSP.
- NSEL: NSAP selector, sometimes also called SEL. It identifies a network service user, and is similar to the TCP/UDP port for upper-layer service in the IP protocol. In IS-IS, SEL is usually set to 00 to imply the routing device.
- NSAP: Network Service Access Point is the complete address for CLNS packets, including OSI address and high-level process, with structure containing area ID, System ID and SEL. A NSAP address with SEL being 00 implies the NET entity. It is similar to the combination of IP address and IP protocol number.
- SNPA: Sub-network Point of Attachment provides physical link and network layer services, and is similar to the MAC address in IP and DLCI, WAN and HDLC in FR.
- L1 router: The router inside an area. It only accepts relevant information from the local area. In order to reach other areas, a default route to the closest L2 must be saved in L1.
- L2 router: the trunk router between different areas. L1 cannot be directly connected with L2.
- L1/L2 router: The boundary router used to connected L1 router and L2 router, containing the databases of both L1 router and L2 routers. It is similar to the ABR in OSPF.
- Pseudonode: The identifier of broadcast subnet of LAN. Pseudonode makes broadcast media a virtual routing device, while every router acts as its interface. DIS manages the adjacency between router and pseudonode.
- NET: network entity title, a part of OSI address describing the area and system ID.
- Circuit: Circuit is the term for interface in IS-IS. NSAP and NET represent the entire device, while circuit represents the interface. For a point-to-point interface, the circuit ID is 1 byte long. For example, the circuit ID is 0x00 in HDLC; in a broadcast network such as LAN, the circuit ID is generally 7 bytes long combining the System ID, such as 1921.6800.0001.01.

To learn more details about IS-IS, please refer to ISO 10589 and RFC 1195.

Configuring IS-IS

The default configurations of IS-IS are given below:

Function	**Default setting**
Network interface	Interface metric: 10 Advertised Hello interval: 10 seconds Advertised CSNP interval: 10 seconds Minimal interval for LSP transmission: 33 milliseconds LSP retransmission interval: 5 seconds Hello multiplier number: 3 Priority for routing node election: 64 Circuit type: Level-1-2 Authentication password: none
System type	Level-1-2
Default route	Level-1 default route: enabled Level-2 default route: disabled.
LSP authentication password	NA
IS-IS GR	IS-IS GR Restarter: Disabled IS-IS GR Helper: Enabled
Summary-address	Undefined
Overload flag bit	Undefined
LSP checksum error	Enabled
Adjacency change logging	Disabled
LSP refresh interval	900 seconds
LSP lifetime	1200 seconds

Enabling IS-IS

Unlike other routing protocols, you need to first create an IS-IS routing process and specify the interfaces on which IS-IS shall be enabled.

Creating IS-IS Routing Process

Command	Function
Ruijie(config)# **router isis** [*tag*]	Starts IS-IS routing process, with tag being the name of IS-IS process.
Ruijie(config-router)# net areaAddress. SystemId.00	Configures the NET address of IS-IS.

- Create IS-IS route process:
- To run IS-IS routing protocol, first create IS-IS routing process in global configuration mode; you can also add "Tag" behind "router isis". This Tag refers to the name of IS-IS routing process. You can also choose not to configure the name of IS-IS routing process. You can configure different IS-IS routing processes by adding different Tags.
- Configure IS-IS protocol's system ID and area address:

- System ID is the only identifier of IS in an autonomous system. Therefore, System ID must be unique in the entire autonomous system. In IS-IS, each area can have one or multiple area addresses, and generally only one area address is needed. Area repartition can be done by configuring multiple area addresses. When configuring multiple area addresses for one IS, the System ID must be identical.

Example:

```
Ruijie(config-router)# net 49.0001.0000.0000.0001.00
```

In the above configuration command, the area address is 49.0001 and System ID is 0000.0000.0001.Dots in the numbers are for your convenience only.

Caution

Level-1 IS routing nodes in the same area must be configured with the same area address. Currently, the core routing table will not be sensitive to the IS-IS process generating the routing table.

Caution

By default, CPU protection is enabled on the switch. For packets corresponding to each destination group address of IS-IS (AllISSystems, AllL1ISSystems, AllL2ISSystems), there will be default limit in number when sent to the CPU. For example, the default limit is 400pps. If there are many adjacencies or if Hello packets are sent at short intervals, the IS-IS packets received by the switch may exceed the default limit, leading to the continual oscillation of adjacencies. In such a case, the limit for IS-IS packets must be raised by configuring such global commands of **cpu-protect type isis-is pps, cpu-protect type isis-l1is pps and cpu-protect type isis-l2is pps**.

Configuring IS-IS Protocol on the Interface

After global IS-IS protocol is enabled, you need to configure IS-IS protocol on the interface.

Use the following command to configure IS-IS protocol on the interface.

Command	Function
Ruijie(config-if)# ip router isis [tag]	Enables IPv4 IS-IS on the specified interface, with "tag" being the name of IS-IS process.
Ruijie(config-if)# **ipv6 router isis** [*tag*]	Enables IPv6 IS-IS on the specified interface, with "tag" being the name of IS-IS process.

Caution

When configuring IP address, the IP address must be in the same network segment as the IP address of adjacent interface.

Caution If the IP address is not in the same network segment as the IP address of adjacent interface, the adjacency cannot be established.

Caution If the interface needs to join the specified IS-IS process, the Tag name of this IS-IS process must be added after "ip router isis".

Caution By configuring the no ip routing command in global configuration mode, IS-IS will disable IPv4 routing function on all interfaces, namely "no ip router isis [tag]" will be executed automatically on all interfaces, while other IS-IS configurations will remain unchanged.

Caution When you configure the IPv6 address, the local link address will be configured by default

Caution If the IPv6 address or the IPv6 address of the adjacent interface does not have the local link address, the adjacency cannot be established.

Caution If you want to add the interface to the designated IS-IS process, attach the Tag of this IS-IS process to the end of **ipv6 router isis.**

Caution If the **no ipv6 unicast-routing** command is executed in global configuration mode, IS-IS will disable IPv6 routing on all interfaces. Namely, the **no ipv6 router isis** [*tag*] command is executed automatically on all interfaces while other IS-IS configurations remain unchanged.

Caution In order to avoid routing blackholes on the network where IPv4 and IPv6 coexist, if protocols supported by two devices or interfaces are not the same, adjacency will not be set up. In this case, please check whether the network topology has any problem. If there is no problem with the network topology and there are no routing blackholes, configure different instances to perform IPv4 and IPv6 routes learning.

Configuring IS-IS Hello Packets

Configuring the Advertised Hello Interval

IS-IS will periodically send Hello packets on the interface, while routing deviceswill discover and maintain adjacencies through the reception and sending of Hello packets. Complete the following configuration in interface configuration mode to set the Hello packet broadcast interval:

Command	Function

Ruijie(config-if)#**isis hello-interval** { **interval** | **minimal** } [**level-1** | **level-2**]	Configures the interval for sending Hello packets on the interface, in the range of 1 to 65535 seconds.

Use the command to change the interval for sending Hello packets. DIS in broadcast network will send Hello packets at an interval which is three times shorter than non-DIS. If IS is elected as DIS on this interface, the interface will send Hello packets every 3.3 seconds by default.

If the key word minimal is used, then the holdtime in Hello packets will be set to 1, and hello interval will be calculated based on the hello-multiplier. For example, if hello-multiplier is configured to 3 and the isis hello-interval minimal command is configured at the same time, the value of hello-interval shall be 1s/3 (333ms).

Caution

By default, CPU protection is enabled on the switch. For packets corresponding to each destination group address of IS-IS (AllISSystems, AllL1ISSystems, AllL2ISSystems), there will be default limit in number when sent to the CPU. For example, the default limit is 400pps. If there are many adjacencies or if Hello packets are sent at short intervals, the IS-IS packets received by the switch may exceed the default limit, leading to the continual oscillation of adjacencies. In such a case, the limit for IS-IS packets must be raised by configuring such global commands of cpu-protect type isis-is pps, cpu-protect type isis-l1is pps and cpu-protect type isis-l2is pps.

Configuring Hello Multiplier Number

IS-IS will periodically send the Hello packet on the interface, and advertise the adjacency hold time of the IS device in the header of the Hello packet. Neighbors will update adjacencies based on the holdtime field in the Hello packet header. The holdtime value in the Hello packet header equals to the hello-interval value multiplies the hello-multiplier value.

Complete the following configuration in interface configuration mode to set the Hello packet holdtime multiplier:

Command	**Function**
Ruijie(config-if)# **isis hello-multiplier multiplier-number** [**level-1** | **level-2**]	Configures Hello multiplier number on the interface.

Use the command to change the holdtime multiplier of Hello packets and the holdtime. The holdtime of Hello packets can also be changed by changing hello-interval or changing both of them.

Configuring Hello packet failure number

IS-IS protocol maintains relationships with adjacent routers by sending and receiving Hello packets. When the local router fails to receive a specific number of Hello packets from peers continuously, adjacent routers will be considered failed. The default number is three. Complete the following configuration in interface configuration mode to set the failure number of Hello packets:

Command	**Function**

Ruijie(config-if)# **isis hello-multiplier** *multiplier-number* [**level-1** | **level-2**]	Sets the number of failure Hello packets on the interface.

Configuring IS-IS LSP

Configuring LSP Minimal Transmission Interval

Complete the following configuration to set the minimum interval for sending LSP packets continuously by IS-IS on the interface:

Command	**Function**
Ruijie(config-if)# **isis lsp-interval** *interval*	Configures the minimal interval (1-4294967295 milliseconds) for sending LSPs on the interface.

Configuring LSP Retransmission Interval

On a point-to-point link, if the local router fails to receive any reply after sending LSPs for a while, it will assume that the LSPs sent formerly are lost or discarded. To ensure the reliability the LSP sending, the local routing device will retransmit the same LSPs. Complete the following configuration in interface configuration mode to set the packets retransmission interval:

Command	**Function**
Ruijie(config-if)# **isis retransmit-interval** *interval*	Configures the interval (1-65535 seconds) for retransmit LSPs on the point-to-point link.

Configuring LSP Refresh Interval

To ensure that each network node can maintain the latest LSP, LSP will periodically refresh the current LSP, and such interval is called LSP refresh interval. With this mechanism, LSPs can remain synchronized in the entire area. Complete the following configuration in IS-IS protocol configuration mode to set the LSP refreshing frequency:

Command	**Function**
Ruijie(config-router)# **lsp-refresh-interval** *interval*	Configures LSP refresh interval (1-65535 seconds).

Caution The lsp-refresh-interval shall be less than the max-lsp-lifetime.

Configuring LSP Lifetime

In LSP, there is a field value called LSP lifetime. When the routing device generates LSP, it will set the maximal lifetime in the field for this LSP. When the LSP is received by another routing decvice, the lifetime will decrease gradually, and the old LSP will be replaced if a new LSP is received. If no refreshed LSP is received and the LSP lifetime has decreased to 0, it will still be kept in the link-state database for 60 seconds. If no refreshed LSP is received within the 60 seconds, this LSP will be deleted from LSDB. With this mechanism, LSPs can remain synchronized in the entire area. Complete the following configuration in IS-IS protocol configuration mode to set the lifetime of LSPs generated by the router:

Command	Function
Ruijie(config-router)#**max-lsp-lifetime value**	Configures LSP lifetime (1-65535 seconds).

Caution The max-lsp-lifetime must be greater than lsp-refresh-interval.

Configuring LSP Fragments Expansion

The LSP fragments expansion function is enabled by setting additional system ID and enabling fragments expansion . Execute the following commands in IS-IS routing process configuration mode:

Command	Function
Ruijie(config-router)# **lsp-fragment-extend** [**level-1 | level-2**] [**compatible rfc3786**]	Enables fragments extension.
Ruijie(config-router)#**virtual-system** *system-id*	Sets additional system ID.

Configuring IS-IS SNP

Configuring the Advertised CSNP Interval

Complete Sequence Number PDUs (CSNP) are packets sent by DIS in the broadcast network to maintain link-state database synchronization. CSNPs are also periodic broadcast packets. Complete the following configuration in interface configuration mode to set the CSNP packet broadcasting interval:

Command	Function
Ruijie(config-if)# **isis csnp-interval** *interval* [**level-1** | **level-2**]	Configures the interval (0-65535 seconds) for sending CSNP packets on the interface.

Configure this command to change the interval for sending CSNP packets. By default, DIS on the broadcast network will send CSNP packets every 10 seconds.

For P2P interface network, CSNP packets will only be sent when adjacency is just established; if the interface is set mesh-groups, you can set sending CSNP packets periodically.

If csnp-interval is set to 0, no CSNP will be sent.

If **mesh-group** is required on the IS-IS interface, you need to configure a non-zero interval for sending VSNP packets to synchronize LSP by using the **isis csnp-interval** command so as to ensure complete LSP synchronization among adjacencies in the network.

Configuring IS-IS Level Type

IS-IS protocol supports two-level hierarchy, so as to manage route selection and achieve expandable route selection. Every level only maintains the topology structure of local area.

You can execute "is-type" command in IS-IS router configuration mode to configure IS-IS Level, or execute "circuit-type" command in the interface configuration mode to configure IS-IS Level of this interface. The default is-type and circuit-type are Level-1-2. If these two commands are configured simultaneously, the corresponding interface will only send Level PDUs with is-type being same as circuit-type.

Configuring System Type

You can configure the level of existing routing devices, which can be divided into Level-1 router (intra-area routing device), Level-2 (inter-area routing device) and Level-1-2 router (both an intra-area routing device and an inter-area routing device). If is-type is configured to Level-1 or Level-2-only, the IS-IS process will only process data at this level. Complete the following configuration in IS-IS protocol configuration mode to set Level of the router:

Command	Function
Ruijie(config-router)# **is-type** { **level-1** | **level-1-2** | **level-2-only** }	Configures system type.

Configuring the Interface Circuit Type

You can configure the type of the interface circuit. Complete the following configuration in interface configuration mode to set the type of the interface circuit:

Command	Function
Ruijie(config-if)# **isis circuit-type** { **level-1** | **level-1-2** | **level-2-only** }	Configures the interface circuit type.

If the circuit-type of "Level-1" or "Level-2-only" is configured, then IS-IS will only send PDUs of the same level.

Configuring IS-IS Authentication

You can configure IS-IS authentication to improve the security of IS-IS network. You can configure authentication for IS-IS in the different level ranges, which include IS-IS interface, IS-IS area and IS-IS route domain.

The interface authentication functions during adjacency formation. If two IS-IS devices are configured with different interface authentication passwords, the adjacency won't be formed, thus avoiding unauthorized or unauthenticated IS-IS devices from joining an IS-IS network in which authentication is required. The interface authentication password is encapsulated in the Hello packets.

IS-IS domain authentication and routing domain authentication are used to authenticate LSP, CSNP and PSNP packets, so as to avoid unauthorized or unauthenticated routing information from entering IS-IS link-state database. The authentication password is encapsulated in the corresponding LSP, CSNP and PSNP packets.

Currently, the following two kinds of authentication methods are provided: plain text authentication and MD5 authentication. The approach of plain text authentication can only guarantee limited security as the password carried by packets can be seen directly. The approach of MD5 authentication will provide better security as the password carried by packets has been encrypted using MD5 algorithm.

Configuring Interface Authentication

Configuring Interface Plain-text authentication

You can configure plain-text authentication password for IS-IS interface. The authentication password will be encapsulated in Hello packets sent on the interface; when Hello packets are received, consistency of the password will be examined.

IS-IS interface authentication should be configured in interface configuration mode. You can use the following command to configure interface plain-text authentication:

Command	**Function**
Ruijie(config)# **interface** *interface-name* Ruijie(config-if)# **isis password** *password* [**send-only**] [**level-1** | **level-2**]	Configures plain-text authentication password for Hello packets transmitted on the interface. When the send-only is specified, the authentication password is only applicable to the sent Hello packets rather than the received Hello packets. When Level is not specified, the authentication password configured is by default applicable to every Level. When configuring this command, if "isis authentication mode" has been executed, this command won't be configured successfully. Both commands can be used to configure IS-IS interface authentication, but this command has a lower priority level. To configure this command, you need to delete "isis authentication mode" command first.

You can also use the following commands to configure plain-text authentication for IS-IS interface:

Command	**Function**
Ruijie(config-if)# **isis authentication mode text** [**level-1** | **level-2**]	Use this command to specify the mode of IS-IS interface authentication. When Level is not specified, the authentication mode configured is by default applicable to every Level. When configuring this command, if "isis password password [level-1|level-2]" has been executed, the said command will be overwritten by this command. Both commands can be used to configure IS-IS interface authentication, but this command has a higher priority level.

Ruijie(config-if)# **isis authentication key-chain name-of-chain** [**level-1** | **level-2**]	Configures the key chain used by IS-IS interface authentication. When Level is not specified, the key chain configured is by default applicable to every Level. This command must be configured together with "isis authentication mode" command in order to achieve IS-IS interface authentication. Neither of them can be omitted.
Ruijie(config-if)# **isis authentication send-only** [**level-1** | **level-2**]	(Optional) The IS-IS interface authentication can only apply to the packets sent. No authentication will be performed on packets received. When Level is not specified, the send-only authentication mode configured is by default applicable to every Level. This command can be used to avoid network oscillation caused by the failure in temporary authentication. Before deploying IS-IS authentication for the entire network, configure this command for all devices; after configuring the aforementioned two commands for all devices, execute "no isis authentication send-only" command to restore the authentication of packets received, so as to carry out smooth authentication deployment and avoid network oscillation.

Configuring Interface Encryption Authentication

You can configure encryption authentication password for IS-IS interface. The authentication password will be encapsulated in Hello packets sent on the interface; when Hello packets are received, consistency of the password will be examined.

IS-IS interface authentication should be configured in interface configuration mode. You can use the following commands to configure interface encryption authentication:

Command	Function
Ruijie(config-if)# **isis authentication mode md5** [**level-1** | **level-2**]	Sets the IS-IS interface authentication mode. When Level is not specified, the set authentication mode applies to all Levels. If the isis password password [level-1 | level-2] command is previously configured, it will be covered by this command. The two commands can both configure IS-IS interface authentication. This command has a higher priority.
Ruijie(config-if)# **isis authentication key-chain** *name-of-chain* [**level-1** | **level-2**]	Sets key-chain used for IS-IS interface authentication. When Level is not specified, the set key-chain applies to all Levels. This command must be used together with the isis authentication mode command to authenticate IS-IS interface.

Ruijie(config-if)# **isis authentication send-only** [**level-1** | **level-2**]	(Optional) The IS-IS interface authentication can only apply to the packets sent. No authentication will be performed on packets received. When Level is not specified, the set authentication and password apply to all Levels. This command can be used to avoid network oscillation caused by the failure in temporary authentication during configuration of IS-IS authentication. Before deploying IS-IS authentication for the entire network, configure this command for all devices; after configuring the aforementioned two commands for all devices, use no isis authentication send-only command to restore the authentication of packets received, so as to carry out smooth authentication deployment and avoid network oscillation.

Configuring Area Authentication

Configuring Area Plain-text Authentication

You can configure plain-text authentication password for IS-IS area. The authentication password will be encapsulated in LSP, CSNP and PSNP packets in the area (Level-1); when the packets are received, consistency of the password will be examined.

IS-IS area authentication must be configured in IS-IS process mode. You can use the following command to configure IS-IS area plain-text authentication:

Command	Function
Ruijie(config-router)# **area-password** *password* [**send-only**]	Configures area (Level-1) plain-text authentication password. When send-only is specified, the authentication password is only applicable to the sent packets rather than the received packets. When configuring this command, if "authentication mode" has been executed, this command won't be configured successfully. Both commands can be used to configure IS-IS area authentication, but this command has a lower priority level. To configure this command, you need to delete "authentication mode" command first.

You can also use the following commands to configure plain-text authentication for IS-IS area:

Command	Function

Ruijie(config-router)# **authentication mode text level-1**	Use this command to specify the mode of IS-IS area authentication. When configuring this command, if "area-password password" has been executed, the said command will be overwritten by this command. Both commands can be used to configure IS-IS area authentication, but this command has a higher priority level.
Ruijie(config-router)# **authentication key-chain** *name-of-chain* **level-1**	Configures the key chain used by IS-IS area authentication. This command must be configured together with "authentication mode" command in order to perform IS-IS area authentication. Neither of them can be omitted.
Ruijie(config-router)# **authentication send-only level-1**	(Optional) The IS-IS area authentication can only apply to the packets sent. No authentication will be performed on packets received. This command can be used to avoid network oscillation caused by the failure in temporary authentication. Before deploying IS-IS authentication for the entire area, configure this command for all devices; after configuring the aforementioned two commands for all devices, execute "no authentication send-only" command to restore the authentication of packets received, so as to carry out smooth authentication deployment and avoid network oscillation.

Configuring Area Encryption Authentication

You can configure encryption authentication password for IS-IS area. The authentication password will be encapsulated in LSP, CSNP and PSNP packets in the area (Level-1); when the packets are received, consistency of the password will be examined.

IS-IS area authentication must be configured in IS-IS process mode. You can use the following commands to configure IS-IS area encryption authentication:

Command	Function
Ruijie(config-router)# **authentication mode md5 level-1**	Sets the IS-IS area authentication mode. If the area-password password command is previously configured, it will be covered by this command. The two commands can both configure IS-IS area authentication. This command has a higher priority.

Ruijie(config-router)# **authentication key-chain** *name-of-chain* **level-1**	Sets key-chain used for IS-IS area authentication. This command must be used together with the authentication mode command to authenticate IS-IS area authentication.
Ruijie(config-router)# **authentication send-only level-1**	(Optional) The IS-IS area authentication can only apply to packets sent. No authentication will be performed on packets received. This command can be used to avoid network oscillation caused by the failure in temporary authentication during configuration of IS-IS authentication. Before deploying IS-IS authentication for the entire area, configure this command for all devices; after configuring the aforementioned two commands for all devices, use no authentication send-only command to restore the authentication of packets received, so as to carry out smooth authentication deployment and avoid network oscillation.

Configuring the Routing Domain Authentication

Configuring Routing Domain Plain-text Authentication

You can configure plain-text authentication password for IS-IS routing domain. The authentication password will be encapsulated in LSP, CSNP and PSNP packets in Level-2; when the packets are received, consistency of the password will be examined.

IS-IS routing domain authentication must be configured in IS-IS process mode. You can use the following command to configure IS-IS routing domain plain-text authentication:

Command	**Function**
Ruijie(config-router)# domain-password password [send-only]	Configure routing domain (Level-2) plain-text authentication password. When send-only is specified, the authentication password is only applicable to the sent packets rather than the received packets. When configuring this command, if "authentication mode" has been executed, this command won't be configured successfully. Both commands can be used to configure IS-IS routing domain authentication, but this command has a lower priority level. To configure domain-password, you need to delete "authentication mode" command first.

You can also use the following commands to configure plain-text authentication for IS-IS routing domain:

Command	**Function**
Ruijie(config-router)# **authentication mode text level-2**	Use this command to specify the mode of IS-IS routing domain authentication. When configuring this command, if "domain-password password" has been executed, the said command will be overwritten by this command. Both commands can be used to configure IS-IS routing domain authentication, but this command has a higher priority level.
Ruijie(config-router)# **authentication key-chain** *name-of-chain* **level-2**	Configures the key chain used by IS-IS routing domain authentication. This command must be configured together with authentication mode command in order to achieve IS-IS routing domain authentication. Neither of them can be omitted.
Ruijie(config-router)# **authentication send-only level-2**	(Optional) The IS-IS routing domain authentication can only apply to the packets sent. Packets received will not be authenticated. This command can be used to avoid network oscillation caused by temporary authentication failure. Before deploying IS-IS authentication for the entire routing domain, configure this command for all devices; after configuring the aforementioned two commands for all devices, execute no authentication send-only command to restore the authentication of packets received, so as to carry out smooth authentication deployment and avoid network oscillation.

Configuring Routing Domain Encryption Authentication

You can configure encryption authentication password for IS-IS routing domain. The authentication password will be encapsulated in LSP, CSNP and PSNP packets in Level-2; when the packets are received, consistency of the password will be examined.

IS-IS routing domain authentication must be configured in IS-IS process mode. You can use the following commands to configure IS-IS routing domain encryption authentication:

Command	**Function**
Ruijie(config-router)# **authentication mode md5 level-2**	Sets the IS-IS routing domain authentication mode. If the domain-password password command is previously configured, it will be covered by this command. The two commands can both configure IS-IS routing domain authentication. This command has a higher priority.

Ruijie(config-router)# **authentication key-chain** *name-of-chain* **level-2**	Sets key-chain used for IS-IS routing domain authentication. This command must be used together with the authentication mode command to authenticate IS-IS routing domain authentication.
Ruijie(config-router)# **authentication send-only level-2**	(Optional) The IS-IS routing domain authentication can only apply to packets sent. No authentication will be performed on packets received. This command can be used to avoid network oscillation caused by the failure in temporary authentication during configuration of IS-IS authentication. Before deploying IS-IS authentication for the entire routing domain, configure this command for all devices; after configuring the aforementioned two commands for all devices, use no authentication send-only command to restore the authentication of packets received, so as to carry out smooth authentication deployment and avoid network oscillation.

Configuring IS-IS GR

IS-IS Graceful Restart (IS-IS GR) guarantees continuous data forwarding during the process of protocol restart. Currently, the high-end products of Ruijie can support IS-IS GR during main/standby switchover, so as to guarantee continuity of key services.

Operating Mechanism of IS-IS GR

- IS-IS GR Realization Standard

RFC5306: Restart Signaling for IS-IS

- RFC5306 operating mechanism

RFC5306 defines requirements, operating methods and issues to be noticed when executing GR; successful GR depends on two principles: 1) the network topology maintains stable; 2) the node maintains non-stop data forwarding during IS-IS protocol restart.

There are two roles in GR: Restarter and Helper. Accordingly, IS-IS GR can be functionally divided into IS-IS GR Restart Capability and IS-IS GR Help Capability. Devices with GR Restart Capability can send GR requests and proactively execute graceful restart, while devices with GR Help Capability can receive GR requests and help the neighbor to execute graceful restart. The process of GR starts with the sending of GR requests by Restarter. The neighbor devices will enter Help mode after receiving such GR requests and assist Restarter to rebuild link-state database while maintaining the adjacencies with Restarter. The main operating mechanism is shown below:

When proceeding with IS-IS GR, the device will advertise its neighbors to maintain their adjacencies, so that the other devices on the network will not perceive the network change. The topological relationships will remain unchanged, and the neighbors will not recalculate routes and update the forwarding table. On the other hand, the link-state database will be synchronized and restored under the aid of neighbors, so that routes and forwarding table remain unchanged after GR, ensuring continuity of data forwarding.

During graceful restart of Restarter, the following steps will be involved:

- GR Restarter advertises the GR Helpers of such restart

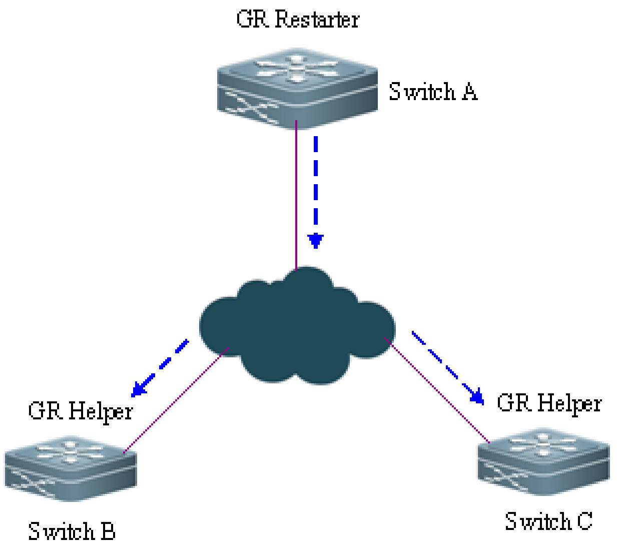


Figure 3 Restart of Restarter by advertising

As shown in Fig 3, Switch A is GR Restarter, Switch B and Switch C are GR Helpers of Switch A. Switch A sends GR requests to all its neighbors, as it needs to maintain the adjacencies during the process of GR. After receiving such requests, all neighbors will maintain the adjacencies with GR Restarter during the GR time (GR grace-period) advertised by the Restarter and send GR replies to Restarter.

- Restart of GR Restarter

As shown in Fig 4, when the GR Restarter proceeds with IS-IS restart, its IS-IS interface will undergo the process from Down to Up. Since the Helper is aware of the protocol restart state of Restarter, it will maintain its adjacency with GR Restarter and the routes acquired from GR Restarter during GR Time.

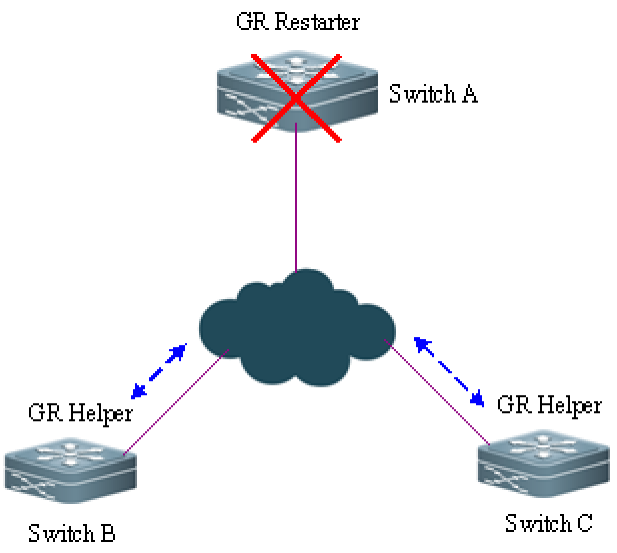


Figure 4 Restart of Restarter

- GR Restarter synchronizes with GR Helper and acquires the topology and routing information

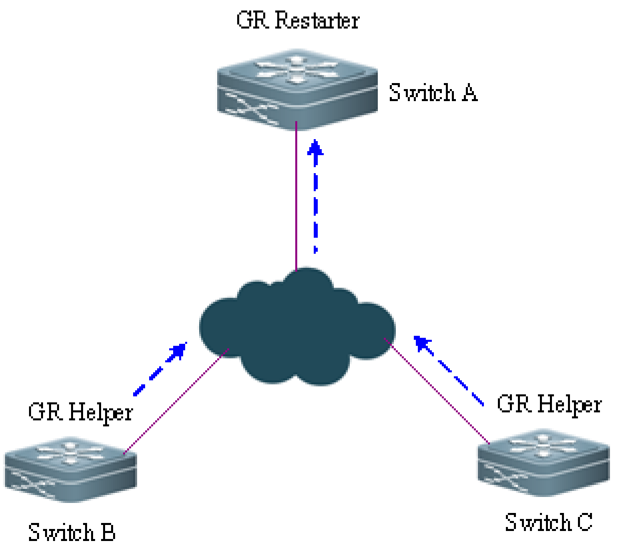


Figure 5 Database synchronization

As shown in Fig 5, after IS-IS protocol restart, GR Restarter will synchronize with GR Helper to acquire topology or routing information, and recalculate its own routing table accordingly. During this process, the forwarding table will not be updated by the routing table.

- GR Restarter completes database synchronization and graceful restart. All devices enter into IS-IS standard protocol interaction state.

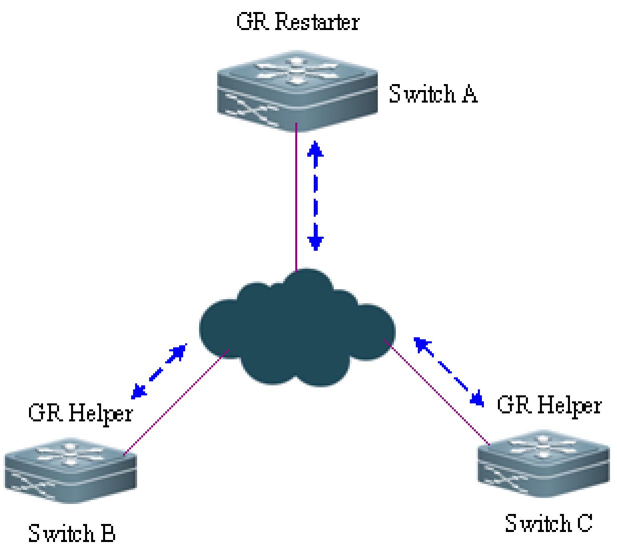


Figure 6 Completion of graceful restart

As shown in Fig 6, Restarter has completed data synchronization and all devices have entered into IS-IS standard protocol interaction state. By this time, the forwarding table will be updated by the routing table of Restarter and invalid entries will be deleted. Since the network maintains stable and Restarter has perfectly restored to the state before restart (completing graceful restart), its routing and forwarding tables will remain unchanged after the restart.

Use of IS-IS GR

GR of routing protocols is usually used to improve the system's reliability in the system that supports separation of the control panel and forwarding panel, thus realizing continued forwarding. IS-IS GR Restart capability depends on products.

When IS-IS GR Restarter capability is enabled, configure the IS-IS adjacency holdtime to no less than 40 seconds on devices with multiple management boards to ensure graceful restart of IS-IS triggered by switch of management boards. It can be achieved by configuring the isis hello-interval and isis hello-multiplier commands. If the holdtime value is less than 40 seconds, the holdtime value in the Hello packet header is set to 40 seconds by default.

The IS-IS GR Help capability only depends on software version. If the software supports IS-IS, the device is equipped with IS-IS GR Help capability.

Configuring IS-IS GR Restarter

To enable IS-IS graceful restart GR Restart capability, you must configure the graceful-restart command to enable graceful restart:

Command	Definition
Ruijie # **configure terminal**	Enters global configuration mode.
Ruijie (config)# **router isis**	Opens IS-IS and enters IS-IS configuration mode.
Ruijie(config-router)# **graceful-restart grace-period seconds**	(Optional) Configures the restart cycle GR Time (default value: 300 seconds).

Ruijie (config-router)# **end**	Returns to privileged mode.
Ruijie # **show isis graceful-restart**	Verifies the configuration.
Ruijie # **write**	(Optional) Save the configuration.

Configuring IS-IS GR Helper

IS-IS GR Help capability is enabled by default. You can also disable GR Help. The following example shows how to disable GR Help capability and re-enable it:

Command	Definition
Ruijie # **configure terminal**	Enters global configuration mode.
Ruijie (config)# **router isis**	Opens IS-IS and enters IS-IS configuration mode.
Ruijie(config-router)# **graceful-restart helper disable**	Disables IS-IS GR Restarter capability on the neighbor of Restarter. The capability is enabled by default.
Ruijie (config-router)# **no graceful-restart helper disable**	Re-enables IS-IS GR Help capability and restores it to the default action.
Ruijie (config-router)# **end**	Returns to privileged mode.
Ruijie # **show isis graceful-restart**	Verifies the configuration.
Ruijie # **write**	(Optional) Save the configuration.

Configuring Linkage between IS-IS and BFD

The IS-IS protocol detects neighbors through Hello packets. After BFD detection is enabled with IS-IS, BFD session is established for the UP neighbors to monitor the neighbor status, Once the BFD neighbor is DOWN, IS-IS performs immediate convergence. The convergence time is reduced to 1s from 30s(by default, IS-IS Hello packets sending interval is 10s on a point-to-point network, and the failure time of the neighbor device is three times of the interval, that is, 30s),

In normal cases, BFD send detecting packets to detect link state with intervals in milliseconds. When the link gets abnormal, for example, the link is disconnected, BFD can detect link anomaly quickly and inform IS-IS to delete neighbors and neighbors-reachable information in LSP packets. IS-IS performs routing calculation again to generate new a route, avoiding the abnormal link and achieving fast convergence, With the introduction of new technologies such as Multi-Service Transport Platform (MSTP), link is congestion-prone in peak periods of data communication. In congestion, BFD can detect link anomaly quickly and inform IS-IS to delete neighbors and neighbors-reachable information in LSP packets. Besides, BFD perform the link switch to avoid congestion. As the IS-IS neighbor detects that the interval to send Hello packets is 10s and the timeout period is 30s. When BFD detects anomaly, the router can receive IS-IS and establish IS-IS adjacency relation. The route restores to the congested link and performs BFD detection again. BFD repeats the process of detecting link anomaly and performing link switch, making the route switched to either the congested link or other links and causing congestion.

Anti-congestion is enabled to avoid routing congestion caused by link congestion. Thus in link congestion, the IS-IS neighbor remains but the neighbor-reachable information is deleted in LSP packets. The route is switched to the non-congested link. After the link restores to normal, or rather non-congested, the neighbor-reachable information in LSP packets is restored and the route is switched back, avoiding routing congestion.

When IS-IS enables anti- congestion, both the **bfd all-interfaces** [**anti-congestion**] and the **bfd up-dampening** commands must be configured on the interface. Configuring only one command may cause ineffective anti- congestion or other network anomalies.

Command	Definition
Ruijie(config-router)# **bfd all-interfaces** [**anti-congestion**]	Enables linkage between IS-IS and BFD on all interfaces.
Ruijie(config-if)# **isis bfd** [**disable** | **anti-congestion**]	Enables or disables linkage between IS-IS and BFD on the interface.

Note

The BFD session needs to be set on the interface before configuring IS-IS with BFD.

Note

When the interface is configured with the **bfd up-dampening** command, the **bfd all-interfaces** [**anti-congestion**] command must be enabled if IS-IS is used with BFD on the interface.

Note

The **bfd all-interfaces** [**anti-congestion**] command must be configured together with the **bfd up-dampening** command on the interface.

Note

IP routing may cause inconsistency between the specified interface and the actual outbound interface of BFD packets, therefore the BFD session cannot be established.

Note

If the specified interface is not the actual inbound interface of BFD packets, the BFD session cannot be established.

Configuring IS-IS OVERLOAD

If the OVERLOAD flag in a non-virtual LSP packet, IS-IS neighbors can be notified for not using a local node as a transit device.

Command	Function
Ruijie(config-router)# **set-overload-bit** [**on-startup** *seconds*] [**suppress** { [**interlevel**] [**external**] }] [**level-1** | **level-2**]	Sets the OVERLOAD flag.

The OVERLOAD flag is used in the following three cases:

- Device overload

If the overload (such as memory insufficiency and CPU full load) occurs on a local IS-IS node, the local routing table will be incomplete or contain no resource transit data. At this time, you can set the OVERLOAD flag in the LSP packet to notify the neighbors for not using the local node as a transit device.

In this case, the configuration does not carry the keyword on-startup, and the user must manually set or cancel the OVERLOAD flag. The user must manually cancel the OVERLOAD flag when the local IS-IS node recovers. Otherwise, the local IS-IS node will always stay in the OVERLOAD state.

- Instant black hole

In the scenario described in RFC3277, an instant black hole (instant blocked route) may occur after the IS-IS node is restarted because the IS-IS convergence speed is faster than that of BGP.

In this case, the configuration must carry the keyword on-startup. The IS-IS node automatically set or cancel the OVERLOAD flag based on the configuration. After the on-startup option is selected, an instant black hole occurs on the IS-IS node after it is restarted. Once a new neighbor relationship is created, the LSP packet carrying the OVERLOAD flag is sent for notifying the neighbor that there is an instant black hole on the local node and therefore do not use the local node as a transit device. When the specified period times out, the IS-IS node immediately sends the LSP packet where the OVERLOAD flag is canceled. This LSP packet notifies the neighbor that the instant black hole does not exist on the local node any more and therefore the neighbor can use the local node as a transit device.

- Requiring that the IS-IS node does not forward true data

The user may need the local IS-IS node to access the production network for experiments or other functions instead of forwarding true data on the network. At this time, the user can set the OVERLOAD flag in the LSP packet to notify the neighbors for not using the local node as a transit device.

In this case, the configuration does not carry the keyword on-startup, and the user must manually set or cancel the OVERLOAD flag. The user can configure the suppress function as required to suppress routing information carried in the LSP packet in the OVERLOAD state. If internal and external routes are suppressed, only the local directly-connected route is notified.

Configuring IS-IS VRF

Make sure VRF is configured before you binding the IS-IS instance to a VRF. To configure an IS-ISv6 neighbor, multi-protocol VRF must be configured and IPv6 must be enabled.

⚠ Caution The following restrictions should be noted for IS-IS and VRF bindings: 1. The IS-IS instances bound to a same VRF must be configured with different system IDs. While the IS-IS instances bound to different VRFs can share a system ID. 2. One IS-IS instance can be bound to only one VRF, and one VRF can be bound to multiple IS-IS instances. 3. When the VRF bound to an IS-IS instance changes, all the interfaces associated with the instance and the redistribution in routing process configuration mode will be removed accordingly.

Execute the following command to configure IS-IS VRF:

Command	Function
Ruijie(config-router)# **vrf** *vrf-name*	Binds an IS-IS instance to a VRF.

Configuring IS-IS SNMP

By default, SNMP software can perform MIB on the first IS-IS instance displayed by the system. If you want to perfom MIB on another instance, specify it manually.

Execute the following command in IS-IS routing process configuration mode to bind the instance used for IS-IS MIB operation:

Command	Function
Ruijie(config-router)# **enable mib-binding**	Binds the current instance to perform MIB.

There are 18 types of IS-IS packets. Based on different features, they are divided into several sets and each set includes several types of IS-IS TRAP packets. Enable IS-IS TRAP globally in global configuration mode (with the **snmp-server enable traps isis** command), specify the host receiving TRAP packets, and use this command to specify the types of IS-IS TRAP packets allowed to be sent in IS-IS routing process configuration mode. Then IS-IS packets can be transmitted.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)#**snmp-server enable traps isis**	Enables IS-IS TRAP globally.

Ruijie(config)#**snmp-server host** 10.1.1.1	Configures global SNMP host and receives IS-IS TRAP packets.
Ruijie(config)#**router isis**	Enters IS-IS routing pricess configuration mode
Ruijie(config-router)# **enable traps all**	Allows all IS-IS TRAP packets to be sent to the host 10.1.1.1.

Configuring Other IS-IS Parameters

Configuring IS-IS Interface Metric

You can configure the interface metric, which must be configured in interface configuration mode as follows:

Command	**Function**
Ruijie(config-if)# **isis metric metric** [**level-1**| **level-2**]	Configures the metric for the interface. This value is only effective when metric-style includes narrow mode.
Ruijie(config-if)# **isis wide-metric metric** [**level-1** | **level-2**]	Configures the wide-metric for the interface. This value is only effective when metric-style includes wide mode.

Configuring the Priority Level of the Specified Routing Node

In the broadcast network, IS-IS needs to elect a designed routing node (DIS) among all routing nodes. The designated router will then create Pseudonode and generate Pseudonode LSP. In the broadcast network, the DIS is elected by priority, and the user can configure different priority values for different Levels. You can set different priority for different Levels. Complete the following configuration in interface configuration mode to set router priority for election:

Command	**Function**
Ruijie(config-if)# **isis priority value** [**level-1** | **level-2**]	Configures the priority for designated router election on the interface.

Caution

The no isis priority command is used to restore the default priority no matter whether the parameter is followed. If you want to modify the configured priority, you can either use isis priority command with parameter specified to overwrite the configured command directly, or configure a new parameter after restoring the priority to the default value.

Configuring to Generate a Default Route

By default, L2 routers don't generate a default route. Execute the following command in IS-IS protocol configuration mode to generate a default route:

Command	**Function**

Ruijie(config-router)# **default-information originate** [*route-map map-name*]	Generates a Level-2 default route and publishes through LSP. If the route-map option is specified, the default route can be generated only when the condition in the route-map is matched.

Configuring Convergent Route

You can create a convergent route to represent a group of routes in the routing table. The process is called route convergence. One convergent route can include multiple routes in a Level. The interface metric of the convergent route is the smallest one of all routes. Complete the following configuration in IS-IS protocol configuration mode to set the route convergence:

Command	Function
Ruijie(config-router)#**summary-address ip-address net-mask** [**level-1** | **level-2** | **level-1-2**]	Sets convergent route.

Route convergence in IS-ISv6 protocol should be set in IS-ISv6 protocol configuration mode:

Command	Function
Ruijie(config-router)# **address-family ipv6 unicast** Ruijie(config-router-af)# **summary-prefix** *ipv6-prefix* / *prefix-length* [**level-1** | **level-2** | **level-1-2**]	Sets IS-IS IPv6 convergent route.

Configuring to Ignore LSP Authentication and Verification Errors

When local IS-IS receives LSP packet, the LSP packet needs to be verified and calculated. In addition, the calculation result is compared with the verification in the LSP packet. That is, the received LSP packet needs to be verified and authenticated. By default, if the calculation result is different from the verification in the LSP packet, the LSP packet is discarded and not processed. If you run the ignore-lsp- errors command to ignore the verification error, the LSP packet is processed normally even an error is verified. The configuration to ignore LSP authentication and errors must be performed in IS-IS protocol configuration mode:

Command	Function
Ruijie(config-router)#**ignore-lsp-errors**	Configures to Ignore LSP authentication and verification errors.

Configuring to Open Adjacent Event Output Switch

To log events when the IS-IS adjacency changes, you need to open the adjacent event output switch. Complete the following configuration in IS-IS protocol configuration mode to set the adjacent event output switch:

Command	Function
Ruijie(config-router)# **log-adjacency-changes**	Enable the logging of IS-IS adjacency change.

Configuring Route Redistribution

Route redistribution can redistribute one routing protocol's routing information to another routing protocol. Route redistribution must be configured in IS-IS configuration mode or IS-IS address-family ipv6 mode:

Caution

In case there are IS-IS Level-2 instances, all IS-IS Level-1 routes will by default be automatically redistributed into IS-IS Level-2 in these instances. Of course, you can also disable the redistribution from Level-1 into Level-2 by executing "no redistribute isis [tag] level-1 into level-2". You can also filter the redistributed Level-1 routes by executing "redistribute isis [tag] level-1 into level-2 {distribute-list access-list-name| route-map route-map-name}”.

Caution

2. Configure no redistritbue {bgp | ospf <1-65535> | rip | connected | static} to disable protocol redistribution. If no redistribute is followed by any other parameter, it means that this parameter is restored to the default setting instead of disabling protocol redistribution. For example: no redistribute bgp will disable bgp redistribution, while no redistribute bgp route-map aa will disable route-map aa filtering during redistribution instead of disabling bgp redistribution.

Caution

3. In the old version software developed by some manufacturers, after configuring metric-type as external, the metric of redistributed route will be added by 64 during route calculation, and the route will be selected according to metric value. This has violated the protocol. In actual applications, the external route may have higher priority than the internal route. During the intercommunication with such manufacturers, if this problem exists, relevant configurations of devices can be adjusted (such as metric or metric-type) to ensure internal routes have higher priority than the external routes.

Monitoring and Maintaining IS-IS

View IS-IS's link status database and transmission and reception of various packets and calculation of SPF through the following configuration and operation to confirm maintenance of IS-IS route.

Command	Function
Ruijie# **show isis** [*tag*] **database** [*FLAGS* | *LEVEL* | *LSPID*]	Displays IS-IS link-state database.
Ruijie# **show isis** [*tag*] **neighbors** [**detail**]	Displays IS-IS neighbors.
Ruijie# **show isis** [*tag*] **virtual-neighbors**	Displays IS-IS neighbors in virtual system.
Ruijie# **show isis** [*tag*] **interface** [*interface-type interface-number*]	Displays IS-IS interface information.
Ruijie# **show isis** [*tag*] **topology** [**l1** | **l2** | **level-1** | **level-2**]	Displays IS-IS connection topology.
Ruijie# **show isis** [*tag*] **ipv6 topology** [**l1** | **l2** | **level-1** | **level-2**]	Displays IS-IS IPv6 unicast topology.
Ruijie# **show isis** [*tag*] **counter**	Displays various statistics of IS-IS.
Ruijie# **show isis** [*tag*] **hostname**	Displays the mapping relation between the device hostname and System ID.
Ruijie# **show isis** [*tag*] **mesh-groups**	Displays the mesh group configurations on each interface.
Ruijie# **show isis** [*tag*] **graceful-restart**	Displays IS-IS GR status.
Ruijie# **show isis** [*tag*] **protocol**	Displays the IS-IS protocol.

For detailed explanation of commands, please refer to *IS-IS Commands*.

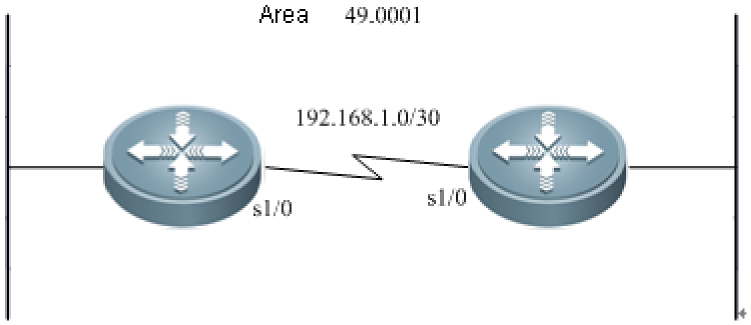
IS-IS Configuration Examples

IS-IS Point-to-Point Serial Link Configuration Example

- Requirement

The connection layout and IP address distribution are shown in Fig 7. A point-to-point network is configured between Device A and Device B.

Figure 7 IS-IS point-to-point serial link configuration



- Detailed configurations

Device A:

Configuring IS-IS routing protocol

```
Ruijie(config)# router isis
Ruijie(config-router)# net 49.0001.0000.0000.0001.00
```

Configuring the Ethernet interface

```
Ruijie(config)# interface GigabitEthernet 0/0
Ruijie(config-if)# ip address 10.1.1.1 255.255.255.0
Ruijie(config-if)# ip router isis
```

Device B:

Configuring IS-IS routing protocol

```
Ruijie(config)# router isis
Ruijie(config-router)# net 49.0001.0000.0000.0002.00
```

Configuring the Ethernet interface

```
Ruijie(config)# interface GigabitEthernet 0/0
Ruijie(config-if)# ip address 10.1.1.2 255.255.255.0
Ruijie(config-if)# ip router isis
```

Device C:

Configuring IS-IS routing protocol

```
Ruijie(config)# router isis
Ruijie(config-router)# net 49.0001.0000.0000.0003.00
```

Configuring the Ethernet port

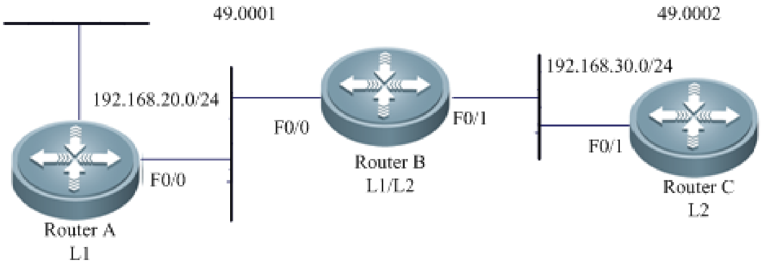
```
Ruijie(config)# interface GigabitEthernet 0/0
Ruijie(config-if)# ip address 10.1.1.3 255.255.255.0
Ruijie(config-if)# ip router isis
```

IS-IS Broadcast Multipoint Link Configuration Example

- Requirement

The connection layout and IP address distribution are shown in Fig 8. Device A, Device B and Device C are interconnected through Ethernet, running the IS-IS routing protocol. Device A is Level-1 node, Device B is Level1-2 node and Device C is Level-2 node. It is required that Hello packets between Device A and Device B, Level-1 LSP and SNP packets adopt plaintext authentication, Hello packets between Device B and Device C, Level-2 LSP and SNP packets adopt MD5 encryption authentication,

Figure 8 IS-IS broadcast multipoint link configuration



- Detailed configurations

Device A:

Configuring IS-IS routing protocol

```
Ruijie(config)# router isis
Ruijie(config-router)# net 49.0001.0000.0000.0001.00
```

Configuring Ethernet interface

```
Ruijie(config)# interface FastEthernet 0/0
Ruijie(config-if)# ip address 10.1.1.1 255.255.255.0
Ruijie(config-if)# ip router isis
```

Device B:

Configuring IS-IS routing protocol

```
Ruijie(config)# router isis
Ruijie(config-router)# net 49.0001.0000.0000.0002.00
```

Configuring Ethernet interface

```
Ruijie(config)# interface FastEthernet 0/0
Ruijie(config-if)# ip address 10.1.1.2 255.255.255.0
Ruijie(config-if)# ip router isis
```

Device C:

Configuring IS-IS routing protocol

```
Ruijie(config)# router isis
Ruijie(config-router)# net 49.0001.0000.0000.0003.00
```

Configuring Ethernet interface

```
Ruijie(config)# interface FastEthernet 0/0
Ruijie(config-if)# ip address 10.1.1.3 255.255.255.0
Ruijie(config-if)# ip router isis
```

IS-IS Authentication Configuration Example

- Requirement

Three devices are interconnected through Ethernet and run IS-IS routing protocol. The connection layout and IP address distribution are shown in Fig 9. Device A is a Level-1 router, Device B is a Level-1-2 router, and Device C is a Level-2 router. Hello packets exchanged between Device A and Device B shall be subject to plain-text authentication, and Level-1 LSP and SNP packets shall be subject to plain-text authentication. Hello packets exchanged between Device B and Device C shall be subject to MD5 encrypted authentication, and Level-2 LSP and SNP packets shall be subject to MD5 encrypted authentication.

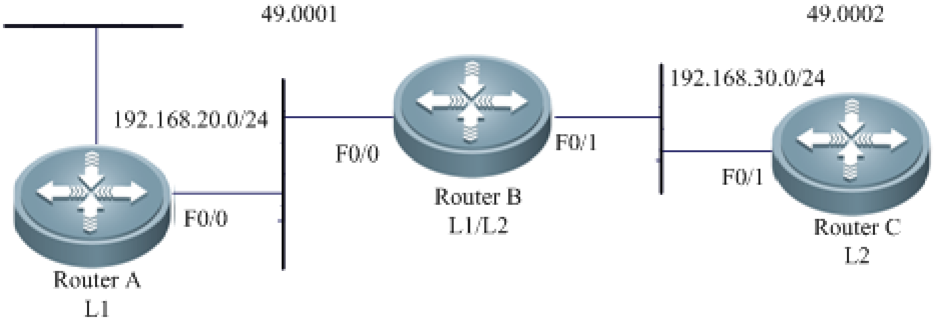


Figure 9 IS-IS authentication configuration

Detailed configurations

Device A:

Configuring IS-IS routing protocol

```
Ruijie(config)# router isis
Ruijie(config-router)# net 49.0001.0000.0000.0001.00
Ruijie(config-router)# is-type level-1
Ruijie(config-router)# area-password aa
```

Configuring Ethernet interface

```
Ruijie(config)# interface FastEthernet 0/0
Ruijie(config-if)# ip address 192.168.20.1 255.255.255.0
Ruijie(config-if)# ip router isis
Ruijie(config-if)# isis password cc
```

Device B:

Configuring the key chain used by IS-IS authentication:

```
Ruijie(config)# key chain kc1
Ruijie(config-keychain)# key 1
Ruijie(config-keychain-key)# key-string aa
Ruijie(config)# key chain kc2
Ruijie(config-keychain)# key 1
Ruijie(config-keychain-key)# key-string bb
Ruijie(config)# key chain kc3
Ruijie(config-keychain)# key 1
Ruijie(config-keychain-key)# key-string cc
```

Configuring IS-IS routing protocol

```
Ruijie(config)# router isis
Ruijie(config-router)# net 49.0001.0000.0000.0002.00
Ruijie(config-router)# authentication mode text level-1
Ruijie(config-router)# authentication key-chain kc1
Ruijie(config-router)# authentication mode md5 level-2
Ruijie(config-router)# authentication key-chain kc2
```

Configuring Ethernet interface

```
Ruijie(config)# interface FastEthernet 0/0
Ruijie(config-if)# ip address 192.168.20.2 255.255.255.0
Ruijie(config-if)# ip router isis
Ruijie(config-if)# isis authentication mode text
Ruijie(config-if)# isis authentication key-chain kc3
Ruijie(config)# interface FastEthernet 0/1
Ruijie(config-if)# ip address 192.168.30.2 255.255.255.0
Ruijie(config-if)# ip router isis
Ruijie(config-if)# isis authentication mode md5
Ruijie(config-if)# isis authentication key-chain kc3
```

Device C:

Configuring the key chain used by IS-IS authentication:

```
Ruijie(config)# key chain kc2
Ruijie(config-keychain)# key 1
Ruijie(config-keychain-key)# key-string bb
Ruijie(config)# key chain kc3
Ruijie(config-keychain)# key 1
Ruijie(config-keychain-key)# key-string cc
```

Configuring IS-IS routing protocol

```
Ruijie(config)# router isis
Ruijie(config-router)# net 49.0002.0000.0000.0002.00
Ruijie(config-router)# is-type level-2
Ruijie(config-router)# authentication mode md5 level-2
Ruijie(config-router)# authentication key-chain kc2
```

Configuring Ethernet interface

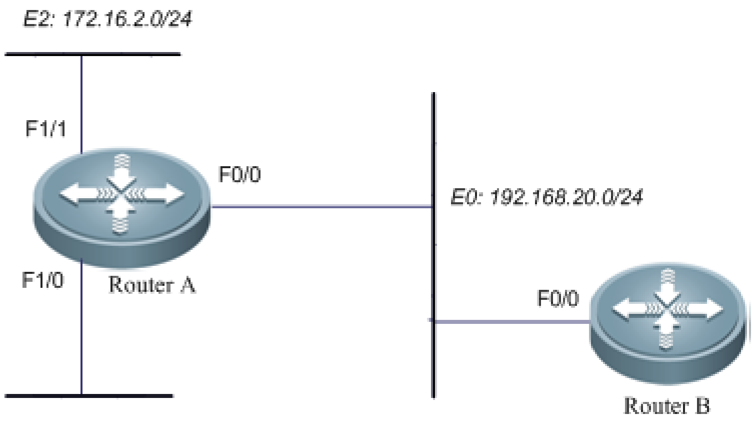
```
Ruijie(config)# interface FastEthernet 0/1
Ruijie(config-if)# ip address 192.168.30.3 255.255.255.0
Ruijie(config-if)# ip router isis
Ruijie(config-if)# isis authentication mode md5
Ruijie(config-if)# isis authentication key-chain kc3
```

IS-IS Route Summary

- Requirement

Two devices are connected through Ethernet. The IP address distribution and device layout are shown in Fig 10.

Figure 10 IS-IS route summary configuration



Requirement

1. Two devices run IS-IS route protocol.

2. Configure Router A, so that Router A only advertises the route of 172.16.0.0/22 instead of routes of 172.16.1.0/24 and 172.16.2.0/24.

■ Detailed configurations

Device A:

Configuring IS-IS routing protocol

```
Ruijie(config)# router isis
Ruijie(config-router)# net 49.0001.0000.0000.0001.00
Ruijie(config-router)# summary-address 172.16.0.0/16 level-1-2
```

Configuring Ethernet interface

```
Ruijie(config)# interface FastEthernet 0/0
Ruijie(config-if)# ip address 192.168.20.1 255.255.255.0
Ruijie(config-if)# ip router isis
Ruijie(config)# interface FastEthernet 1/0
Ruijie(config-if)# ip address 172.16.1.1 255.255.255.0
Ruijie(config-if)# ip router isis
Ruijie(config)# interface FastEthernet 1/1
Ruijie(config-if)# ip address 172.16.2.1 255.255.255.0
Ruijie(config-if)# ip router isis
```

Device B:

Configuring IS-IS routing protocol

```
Ruijie(config)# router isis
Ruijie(config-router)# net 49.0001.0000.0000.0002.00
```

Configuring Ethernet interface

```
Ruijie(config)# interface FastEthernet 0/0
Ruijie(config-if)# ip address 192.168.20.2 255.255.255.0
Ruijie(config-if)# ip router isis
```

Execute show ip route on Device B to see only one summary address:

```
Ruijie(config)# show ip route
i L1   172.16.0.0/16 [115/20] via 192.168.20.1, FastEthernet0/0
```

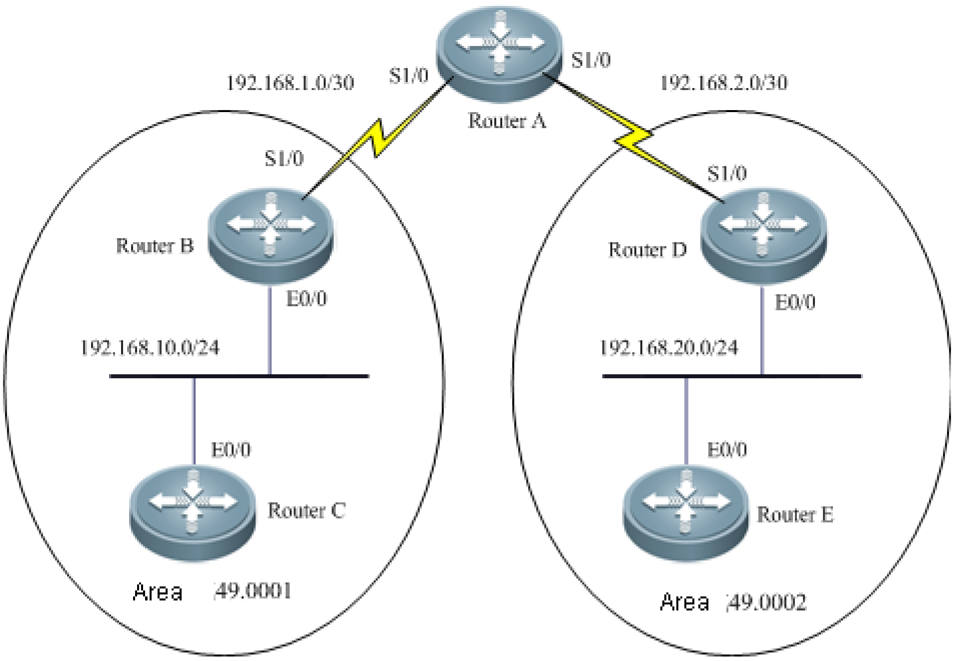
⚠ Caution If Level is no specified when summary-address is used, only Level-2 routes will be summarized by default.

IS-IS Level Configuration Example

- Requirement

See Fig 7 for allocation of IP addresses and connection of devices. P2P serial link connection is deployed between Device A and Device B and C respectively; Ethernet connection is deployed between Device B and Device C; Ethernet connection is deployed between Device D and Device E.

Figure 11 IS-IS level configuration



You need to configure IS-IS area route summary on Router A. Area route summary can only be configured on an area border device.

- Detailed configurations

Device A:

Configuring IS-IS routing protocol

```
Ruijie(config)# router isis
Ruijie(config-router)# net 50.0001.0000.0000.0001.00
Ruijie(config-router)# is-type level-2-only
```

Configuring serial link interface

```
Ruijie(config)# interface Serial 1/0
Ruijie(config-if)# ip address 192.168.1.1 255.255.255.252
Ruijie(config-if)# ip router isis
Ruijie(config)# interface Serial 1/1
Ruijie(config-if)# ip address 192.168.2.1 255.255.255.252
Ruijie(config-if)# ip router isis
```

Device B:

Configuring IS-IS routing protocol

```
Ruijie(config)# router isis
Ruijie(config-router)# net 49.0001.0000.0000.0002.00
```

Configuring Ethernet interface

```
Ruijie(config)# interface GigabitEthernet 0/0
Ruijie(config-if)# ip address 192.168.10.1 255.255.255.0
Ruijie(config-if)# ip router isis
```

Configuring serial link interface

```
Ruijie(config)# interface Serial 1/0
Ruijie(config-if)# ip address 192.168.1.2 255.255.255.252
Ruijie(config-if)# ip router isis
```

Device C:

Configuring IS-IS routing protocol

```
Ruijie(config)# router isis
Ruijie(config-router)# net 49.0001.0000.0000.0003.00
Ruijie(config-router)# is-type level-1
```

Configuring Ethernet interface

```
Ruijie(config)# interface GigabitEthernet 0/0
Ruijie(config-if)# ip address 192.168.10.2 255.255.255.0
Ruijie(config-if)# ip router isis
```

Device D:

Configuring IS-IS routing protocol

```
Ruijie(config)# router isis
Ruijie(config-router)# net 49.0002.0000.0000.0004.00
```

Configuring Ethernet interface

```
Ruijie(config)# interface GigabitEthernet 0/0
Ruijie(config-if)# ip address 192.168.20.1 255.255.255.0
Ruijie(config-if)# ip router isis
```

Configuring serial link interface

```
Ruijie(config)# interface Serial 1/0
Ruijie(config-if)# ip address 192.168.2.2 255.255.255.252
Ruijie(config-if)# ip router isis
```

Device E:

Configuring IS-IS routing protocol

```
Ruijie(config)# router isis
Ruijie(config-router)# net 49.0002.0000.0000.0005.00
Ruijie(config-router)# is-type level-1
```

Configuring Ethernet interface

```
Ruijie(config)# interface GigabitEthernet 0/0
Ruijie(config-if)# ip address 192.168.20.2 255.255.255.0
Ruijie(config-if)# ip router isis
```

IS-ISv6 Simplest Configuration

- Requirement

The connection layout and IPv6 address distribution are shown in Fig 12. Device A and Device B are interconnected through Ethernet.

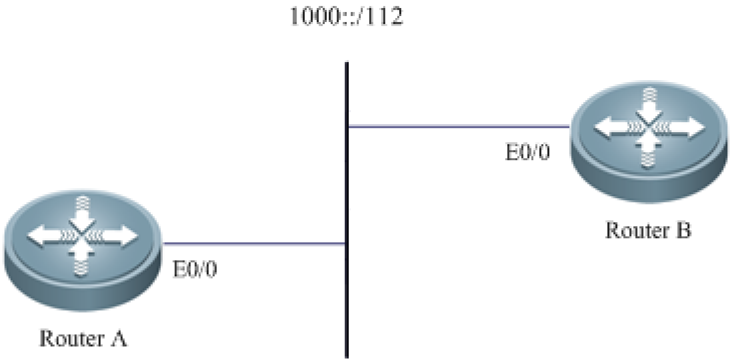


Figure 12 IS-ISv6 configuration

- Detailed configurations

Device A:

Configuring IS-IS routing protocol

```
Ruijie(config)# router isis
Ruijie(config-router)# net 49.0001.0000.0000.0001.00
```

Configuring Ethernet interface

```
Ruijie(config)# interface GigabitEthernet 0/0
Ruijie(config-if)# ipv6 address 1000 ::1/112
Ruijie(config-if)# ipv6 router isis
```

Device B:

Configuring IS-IS routing protocol

```
Ruijie(config)# router isis
Ruijie(config-router)# net 49.0001.0000.0000.0002.00
```

Configuring Ethernet interface

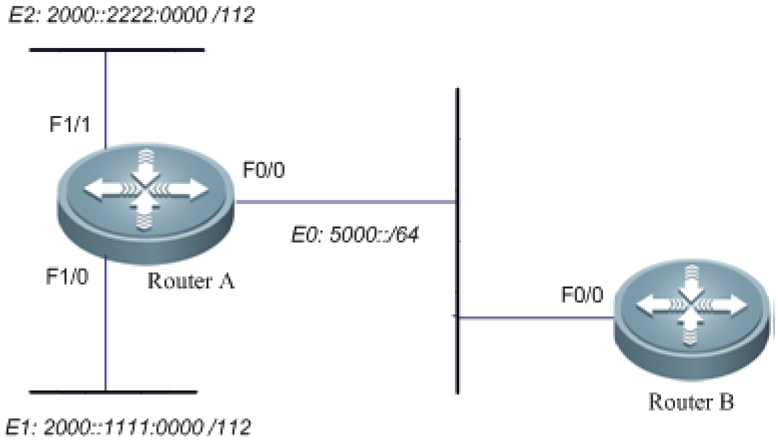
```
Ruijie(config)# interface GigabitEthernet 0/0
Ruijie(config-if)# ipv6 address 1000 ::2/112
Ruijie(config-if)# ipv6 router isis
```

IS-ISv6 Route Summary

- Requirement

Two devices are connected through Ethernet. See Fig 13 for allocation of IP addresses and connection of devices.

Figure 13 IS-ISv6 route summary configuration



Requirement

Two devices run IS-ISv6 route protocol.

Configure Router A, so that Router A only advertises the route of 2000::/96 instead of routes of 2000::1111:0/112 and 2000::2222::0/112.

Detailed configurations

Device A:

```
Configuring IS-IS routing protocol
Ruijie(config)# ipv6 unicast-routing
Ruijie(config)# router isis
Ruijie(config-router)# net 49.0001.0000.0000.0001.00
Ruijie(config-router)# address-family ipv6 unicast
Ruijie (config-router-af)# summary-prefix 2000::/96 level-1-2
Ruijie (config-router-af)# exit-address-family
Configuring Ethernet interface
Ruijie(config)# interface FastEthernet 0/0
Ruijie(config-if)# ipv6 address 5000::1/64
Ruijie(config-if)# ipv6 router isis
Ruijie(config)# interface FastEthernet 1/0
Ruijie(config-if)# ipv6 address 2000::1111:0001/112
Ruijie(config-if)# ipv6 router isis
Ruijie(config)# interface FastEthernet 1/1
Ruijie(config-if)# ipv6 address 2000::2222:0001/112
Ruijie(config-if)# ipv6 router isis
```

Device B:

```
Configuring IS-IS routing protocol
Ruijie(config)# ipv6 unicast-routing
Ruijie(config)# router isis
Ruijie(config-router)# net 49.0001.0000.0000.0002.00
Configuring Ethernet interface
Ruijie(config)# interface FastEthernet 0/0
Ruijie(config-if)# ipv6 address 5000::2/64
Ruijie(config-if)# ipv6 router isis
```

Execute "show ipv6 route" on Device B to see only one summary address:

```
Ruijie(config)# show ipv6 route
I1  2000::/96 [115/20]    via FE80::C800:1BFF:FEF8:1C, FastEthernet1/0
```

Caution

If Level is no specified when summary-prefix is used, only Level-2 routes will be summarized by default.

Configuring NHRP

Overview

Next Hop Resolution Protocol (NHRP) is defined by the IETF in RFC 2332. It is used to enable a source node (a host or router) on a non-broadcast multiple access (NBMA) network to obtain the Internet address and NBMA subnet address of "the next hop" destined to the destination node.

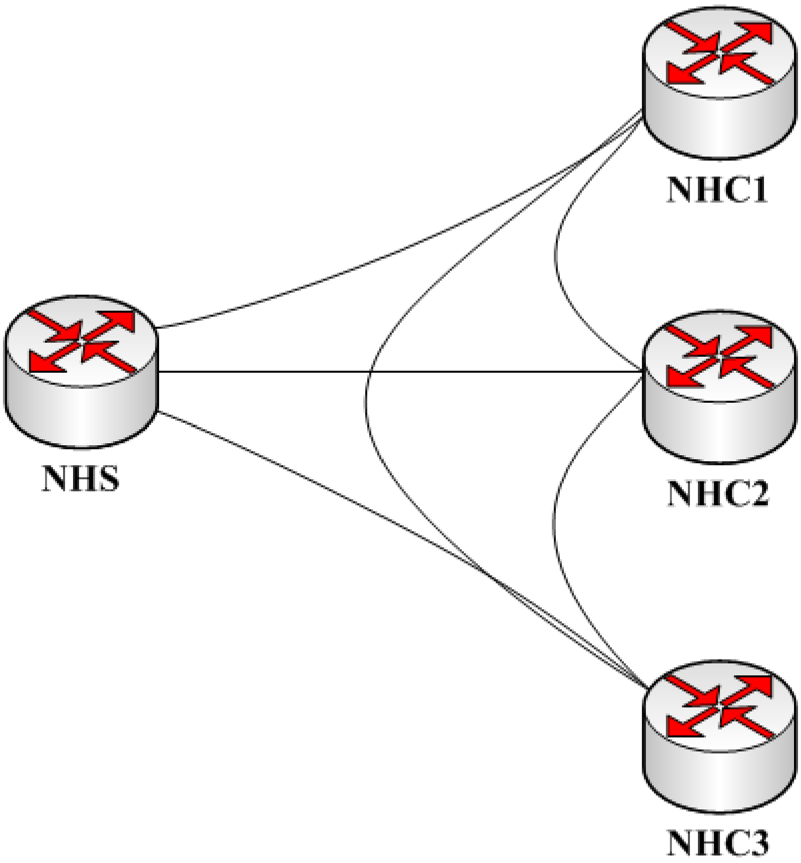
NHRP is a typical client/server protocol and comprises the Next Hop Server (NHS) and Next Hop Clients (NHCs). Each subnet comprises at least one NHS. One NHS can serve multiple subnets and receive request and resolution packets from multiple NHCs. Each end system is an NHC.

Working Principle

NHRP is similar to ARP that is used to complete resolution from IP addresses to MAC addresses. NHRP is used to complete resolution from internal VPN addresses to NBMA addresses (external addresses).

An NHS is used to maintain the NHRP database at the public IP address of each branch NHC. A branch NHC registers the IP address of its external port to the central router NHS through NHRP. The branch NHC can use a dynamic IP address. When the branch NHC wants to establish a direct tunnel with other branch NHCs, the branch NHC queries the NHRP database of the NHS for resolution to determine the real addresses of the other branch NHCs and make redundancy of the addresses to its own NHRP database. In this way, a branch NHC can dynamically join a network, which addresses the problem of establishing VPN tunnels through dynamic IP addresses. In addition, the branch NHC can directly exchange data with other branch NHCs by using the resolved addresses, rather than using the NHS for transmission, finally forming a mesh topology.

Figure 0-1 NHRP network topology



Negotiation packets of NHRP fall into the following types:

- NHRP Resolution Request: NHRP resolution request packet. For packets with uncertain mapping between IP addresses and NBMA addresses, an NHC sends a resolution request packet to the NHS for negotiation and obtain the mapping.
- NHRP Resolution Reply: NHRP resolution reply packet. After receiving the request, the NHS queries its cache table and sends the query result as the reply information to the NHC for address negotiation.
- NHRP Registration Request: NHRP registration request packet. After the NHC joins the network, the NHC needs to register the mapping between its IP address and NBMA address to the NHS in the current network. Therefore, the NHC sends a registration request packet to the NHS and registers itself to the NHS cache table.
- NHRP Registration Reply: NHRP registration reply packet. After receiving the registration request packet, the NHS stores the registration information into its cache table and sends an NHRP registration reply packet based on the registration result.
- NHRP Purge Request: NHRP purge request packet. When either NHRP party wants to purge the cache of a peer, the NHRP party will send an NHRP purge request packet.
- NHRP Purge Reply: NHRP purge reply packet. After receiving the NHRP purge request packet, the peer will send a reply packet to the purge requester based on the cache purge result.
- NHRP Error Indication: NHRP error indication packet. If a receiving error packet is generated during NHRP negotiation, this packet will be send to the original sender.

The two primary work processes of NHRP are the registration process and resolution process.

Registration Process

An NHC needs static cache, which is used to cache the mapping between the IP address and NBMA address of the NHS in the current network. The NHC generates and sends a registration request packet to the NHS. After receiving the packet, the NHS will generate dynamic cache, and generate and send a registration reply packet to the NHC for registration.

Resolution Process

When no NBMA address is found for the mapped IP address, the NHS and NHC will send a resolution request packet for negotiation about a proper address.

As shown in the figure above, HHC1 wants to ping the IP address of NHC2. The resolution negotiation process is as follows:

- When pinging packets, NHC1 queries the NBMA address mapped to the IP address of NHC2. Since no address is resolved, NHC1 sends a resolution request packet to the NHS.
- After receiving the resolution request packet, the NHS sends the corresponding registration cache entry of NHC2 to NHC1 as a resolution reply packet. In this way, NHC1 has learned the NBMA address mapped to the IP address destined to NHC2.
- After the ping packet reaches NHC2, NHC2 will send an ICMP reply packet. Since the source IP address of the ping packet is unresolved, NHC2 also sends a request packet to the NHS to resolve the correct address of NHC1 by following the method of NHC1.
- Then, data packets from NHC1 to NHC2 will be transmitted through the direct route between them, rather than be transmitted through the NHS, forming a mesh network.

If the NHRP resolution request is triggered by a data packet, NHC1 will select one of the following methods to process the data packet when waiting for an NHRP resolution response:

① Releases a data packet.

② Retains the data packet until the NHRP resolution response arrives and another optimum path is available.

③ Sends a data packet along the route.

Method ③ is the recommended default policy since it can resolve subnet-layer addresses as well as send data flows to destination sites.

Protocols and Standards

RFC 2332.

Default Configuration

NHRP is disabled by default.

Configuring NHRP

Enabling NHRP

The NHRP function can take effect on an interface only after it is enabled.

	Command	Description
Step 1	Ruijie(config)#**interface tunnel** *tunnel-number*	Enters a tunnel interface to start NHRP configuration.
Step 2	Ruijie(config-Tunnel *tunnel-number*)#**tunnel mode gre ip**	Configures the tunnel type as GRE. This type of tunnels is used to form a star topology. The tunnel type of the NHS must be MGRE.
	Or:	
	Ruijie(config-Tunnel *tunnel-number*)#**tunnel mode gre multipoint**	Configures the tunnel type as MGRE. This type of tunnels is used to form a mesh topology. The tunnel type of the NHS must be MGRE.
Step 3	Ruijie(config-Tunnel *tunnel-number*)#**ip nhrp network-id***nhrp-number*	Enables the NHRP function on a tunnel interface. *nhrp-number*: Indicates a 32-bit ID. <1-4294967295>.
	Ruijie(config-Tunnel *tunnel-number)*#**no ip nhrp network-id**	Disables the NHRP function.

Caution The NHRP function must be enabled on a tunnel interface and support only GRE and MGRE tunnels.

Example 1: Enable the NHRP function on a tunnel interface.

```
Ruijie(config)#interface tunnel1
Ruijie(config-Tunnel 1)#ip nhrp network-id1
```

Configuring Static IP-Address and NBMA-Address Mapping Cache

The IP-address and NBMA-address mapping cache of NHRP may be either static or dynamic. Static mapping cache can be manually configured whereas dynamic mapping cache is learned by sending protocol packets.

	Command	Description
Step 1	Ruijie(config)#**interface tunnel** *tunnel-number*	Enters a tunnel interface to start NHRP configuration.

	Command	Description
Step 2	Ruijie(config-Tunnel *tunnel-number*)#**ip nhrp map** *ip-address nbma-address*	Configures the IP-address and NBMA-address mapping cache on a tunnel interface. The cache is static and will not time out forever. The former address is an IP address and the latter address is an NBMA address.
	Or:	
	Ruijie(config-Tunnel *tunnel-number*)#**ip nhrp map** *ip-address***mask** *ip-mask nbma-address*	An IP address is a certain address or the IP address of a network segment, which should be configured with a mask.
	Ruijie(config-Tunnel *tunnel-number*)#**no ip nhrp map** *ip-address nbma-address*	Deletes the static configuration.
	Ruijie(config-Tunnel *tunnel-number*)#**no ip nhrp map** *ip-address* **mask***ip-mask nbma-address*	Deletes the static configuration.

Example 1: Configure static NHRP cache without a mask on a tunnel interface.

```
Ruijie(config)#interface tunnel 1
Ruijie(config-Tunnel 1)#ip nhrp map 55.1.1.1 11.1.1.1
```

Example 2: Configure static NHRP cache with a mask on a tunnel interface.

```
Ruijie(config)#interface tunnel 1
Ruijie(config-Tunnel 1)#ip nhrp map 55.1.1.0 mask 255.255.255.0 11.1.1.1
```

Configuring Destination Addresses of Broadcast and Multicast Packets

When a device needs to send broadcast and multicast packets, you need to specify the direction for sending the packets.

	Command	Description
Step 1	Ruijie(config)#**interface tunnel** *tunnel-number*	Enters a tunnel interface to start NHRP configuration.
Step 2	Ruijie(config-Tunnel *tunnel-number*)#**ip nhrp map multicast** *nbma-address*	Configures the NBMA addresses for sending broadcast and multicast packets through the tunnel. Multiple NBMA addresses can be configured.
	Or:	
	Ruijie(config-Tunnel *tunnel-number*)#**ip nhrp map multicast dynamic**	The sending direction can also be dynamically learned. After the dynamic learning command is configured, all NBMA addresses registered on the device can be used as the addresses for sending the broadcast packets.

Command	Description
Ruijie(config-Tunnel *tunnel-number*)#**no ip nhrp map multicast** *nbma-address*	Deletes the static configuration.
Ruijie(config-Tunnel *tunnel-number*)# **no ip nhrp map multicast dynamic**	Disables the configuration.

Caution This command is not supported by point-to-point GRE tunnels, but is supported only by MGRE tunnels.

Example 1: Configure a static address 11.1.1.1 for sending broadcast packets on a tunnel interface.

```
Ruijie(config)#interface tunnel 1
Ruijie(config-Tunnel 1)#ip nhrp map multicast 11.1.1.1
```

Example 2: Configure a dynamic address learning command for sending broadcast packets on a tunnel interface.

```
Ruijie(config)#interface tunnel 1
Ruijie(config-Tunnel 1)#ip nhrp map multicast dynamic
```

Configuring NHS

Configure a static address of an NHS to which a registration request packet needs to be sent on an NHC to trigger sending of the registration request.

	Command	Description
Step 1	Ruijie(config)#**interface tunnel** *tunnel-number*	Enters a tunnel interface to start NHRP configuration.
Step 2	Ruijie(config-Tunnel *tunnel-number*)#**ip nhrp map** *ip-address nbma-address*	Configures the static IP-address and NBMA-address mapping of the NHS.
Step 3	Ruijie(config-Tunnel *tunnel-number*)#**ip nhrp nhs** *nhs-address*	Configures the address of the NHS with which an NHC needs to register for sending a registration request packet.

Ruijie(config-Tunnel *tunnel-number*)#**no ip nhrp nhs** *nhs-address*	Disables the configuration.

Example 1: Configure the NHRP NHS address 11.1.1.1 on a tunnel interface.

```
Ruijie(config)#interface tunnel 1
Ruijie(config-Tunnel 1)#ip nhrp nhs11.1.1.1
```

Configuring Only the NHS Mode

Configure only the NHS mode in which no NHRP registration request packet will be sent.

Command	Description

Step 1	Ruijie(config)#**interface tunnel** *tunnel-number*	Enters a tunnel interface to start NHRP configuration.
Step 2	Ruijie(config-Tunnel *tunnel-number*)#**ip nhrp server-only**	Configures the tunnel interface only in the NHS mode.

Ruijie(config-Tunnel *tunnel-number*)#**no ip nhrp server-only**	Resets the default configuration.

Example 1: Configure only the NHS mode on a tunnel interface.

```
Ruijie(config)#interface tunnel 1
Ruijie(config-Tunnel 1)#ip nhrp server-only
```

Configuring Hold Time of NHRP Cache

Dynamic cache information obtained by NHRP through negotiation has hold time. The following commands are used to configure the hold time of NHRP cache.

	Command	Description
Step 1	Ruijie(config)#**interface tunnel** *tunnel-number*	Enters a tunnel interface to start NHRP configuration.
Step 2	Ruijie(config-Tunnel *tunnel-number*)#**ip nhrp holdtime** *seconds*	Configures the hold time of dynamic NHRP cache. The value of *seconds* ranges from 1 to 65,535. The default value is 7,200 seconds.

Ruijie(config-Tunnel *tunnel-number*)#**no ip nhrp holdtime**	Resets the default configuration.

Example 1: Configure the hold time of NHRP cache as 300 seconds on a tunnel interface.

```
Ruijie(config)#interface tunnel 1
Ruijie(config-Tunnel 1)#ip nhrp holdtime 300
```

Configuring NHRP Authentication

The NHRP information transferred in the same network can be authenticated with the following commands. Only NHRP information that can be successfully authenticated can be processed.

	Command	Description
Step 1	Ruijie(config)#**interface tunnel** *tunnel-number*	Enters a tunnel interface to start NHRP configuration.
Step 2	Ruijie(config-Tunnel *tunnel-number*)#**ip nhrp authentication***string*	Configures the NHRP authentication *string* of 8 bytes, which should be carried by the negotiation packet.
	Ruijie(config-Tunnel *tunnel-number*)#**no ip nhrp authentication**	Disables the configuration.

Example 1: Configure the NHRP authentication string as 123 on a tunnel interface.

```
Ruijie(config)#interface tunnel 1
Ruijie(config-Tunnel 1)#ip nhrp authentication 123
```

Configuring Conditions for NHRP to Trigger a Resolution Request

NHRP can control the conditions for triggering sending of resolution request packets so that certain packets cannot trigger sending of resolution requests.

	Command	Description
Step 1	Ruijie(config)#**access-list***access-list-number*	Configures an ACL rule.
Step 2	Ruijie(config)#**interface tunnel** *tunnel-number*	Enters a tunnel interface to start NHRP configuration.
Step 3	Ruijie(config-Tunnel *tunnel-number*)#**ip nhrp interest** *access-list-number*	Uses the ACL for matching. Negotiation can be performed only for packets matching the ACL. *access-list-number*: Indicates a standard or an extended ACL number, ranging from 1 to 199.
	Or:	
	Ruijie(config-Tunnel *tunnel-number*)#**ip nhrp interest none**	No ACL can be matched and negotiation cannot be triggered.

Step 1	Ruijie(config)#**interface tunnel** *tunnel-number*	Enters a tunnel interface to start NHRP configuration.
Step 2	Ruijie(config-Tunnel *tunnel-number*)#**ip nhrp use***usage-count*	Configures the packet count. NHRP negotiation can be triggered only after *usage-count* (number of packets) packets are sent. The value range is from 1 to 65,535 and the default value is 1.

Ruijie(config-Tunnel *tunnel-number*)#**no ip nhrp interest**	Disables the configuration.
Ruijie(config-Tunnel *tunnel-number*)# **no ip nhrp use**	Resets the default configuration.

Example 1: Configure an NHRP ACL rule on a tunnel interface to match ACL 100.

```
Ruijie(config)#access-list 100 permit ip any any
Ruijie(config)#interface tunnel 1
Ruijie(config-Tunnel 1)#ip nhrp interest 100
```

Example 2: Configure the NHRP delayed triggering command on a tunnel interface. Negotiation can be triggered only after 10 packets are sent.

```
Ruijie(config)#interface tunnel 1
Ruijie(config-Tunnel 1)#ip nhrp use 10
```

Configuring NHRP Redirection

When the central node forwards the mutual access traffic of branch nodes, the central node needs to enable the **ip nhrp redirect** function. After that, the central node will send the **ip nhrp redirect** message to the source branch node to enable the source branch to send an NHRP resolution request.

	Command	Description
Step 1	Ruijie(config)#**interface tunnel** *tunnel-number*	Enters a tunnel interface to start NHRP configuration.
Step 2	Ruijie(config-Tunnel *tunnel-number*)#**ip nhrp redirect**	Enables the NHRP redirection function on the NHS.

Example 1: Enable the **nhrp redirect** function on the tunnel 1 interface.

```
Ruijie(config)#interface tunnel 1
Ruijie(config-Tunnel 1)#ip nhrp redirect
```

Configuring the NHRP Shortcut Function

When forwarding mutual access traffic of branch nodes, the central node identifies whether the inbound and outbound interfaces for forwarding the packets belong to the same DMVPN. If yes, the central node will send a redirection packet to the source branch node. In this case, the source branch node needs to enable the **ip nhrp shortcut** function. The branch node can send a next-hop route resolution request packet only after receiving the redirection packet.

	Command	Description
Step 1	Ruijie(config)#**interface tunnel** *tunnel-number*	Enters a tunnel interface to start NHRP configuration.
Step 2	Ruijie(config-Tunnel *tunnel-number*)#**ip nhrp shortcut**	Enables the NHRP shortcut function on the NHS.

Example 1: Enable the **nhrp shortcut** function on the tunnel 1 interface.

```
Ruijie(config)#interface tunnel 1
Ruijie(config-Tunnel 1)#ip nhrp shortcut
```

Configuring the NHRP Group to Which an NHC Belongs

An NHC can configure an NHRP group to which it belongs and advertises the NHRP group to the NHS during registration so that the NHS executes the per-spoke flow rate-limit function.

	Command	Description
Step 1	Ruijie(config)#**interface tunnel** *tunnel-number*	Enters a tunnel interface to start NHRP configuration.
Step 2	Ruijie(config-Tunnel *tunnel-number*)#**ip nhrp group** *group-name*	Configures the NHRP group to which an NHC belongs.

Example 1: Configure the name of the NHRP group to which an NHC belongs as group-user-1.

```
Ruijie(config)#interface tunnel 1
Ruijie(config-Tunnel 1)#ip nhrp group group-user-1
```

Configuring Per-spoke Flow Rate-Limit on the NHS

You can limit the rates of flows forwarded by the NHS to each NHC. The same flow rate-limit policy will be applied to all NHCs registering to this *group*.

	Command	Description
Step 1	Ruijie(config)#**interface tunnel** *tunnel-number*	Enters a tunnel interface to start NHRP configuration.
Step 2	Ruijie(config-Tunnel *tunnel-number*)#**ip nhrp map group** *group-name***flow-label** *label-value*	Specifies the flow label corresponding to the NHRP group.

Example 1: Configure the flow label for the group named group-user-1 as 3.

```
Ruijie(config)#flow-limit output label 3 300000 3000 3000 conform-action  transmit
exceed-action drop
Ruijie(config)#interface tunnel 1
Ruijie(config-Tunnel 1)#ip nhrp map group group-user-1 flow-label 3
```

Configuring the Transmission Rate of NHRP Negotiation Packets

You can run the following commands to limit the transmission rate of NHRP negotiation packets.

	Command	Description
Step 1	Ruijie(config)#**interface tunnel** *tunnel-number*	Enters a tunnel interface to start NHRP configuration.
Step 2	Ruijie(config-Tunnel *tunnel-number*)#**ip nhrp max-send***pkt-count***every***interval*	Configures the transmission rate of NHRP negotiation packets as *pkt-count* per *interval* seconds. *pkt-count:* Indicates the number of packets sent. The value range is from 1 to 65,535 and the default value is **100**. *interval*: Indicates the interval for sending packets. The value range is from 10 to 65,535 and the default value is **10**.

Ruijie(config-Tunnel *tunnel-number*)#**noip nhrp max-send**	Disables the configuration.

Example 1: Configure the transmission rate of NHRP negotiation packets as 50 per 10 seconds on a tunnel interface.

```
Ruijie(config)#interface tunnel 1
Ruijie(config-Tunnel 1)#ip nhrp max-send 50every10
```

Configuring the Forward and Reverse Record Function

The forward record and reverse record extension fields are carried in NHRP request and reply packets, which are used to record the information about all NHSs before the request and reply packets reach the final NHS. This command is used to enable or disable the record function. The record function is enabled by default. The following commands are used to disable the function.

	Command	Description
Step 1	Ruijie(config)#**interface tunnel** *tunnel-number*	Enters a tunnel interface to start NHRP configuration.
Step 2	Ruijie(config-Tunnel *tunnel-number*)#**no ip nhrp record**	Disables the record function.

Ruijie(config-Tunnel *tunnel-number*)#**ip nhrp record**	Enables the record function again.

Example 1: Disable the record function on a tunnel interface.

```
Ruijie(config)#interface tunnel 1
Ruijie(config-Tunnel 1)#no ip nhrp record
```

Configuring the Address for the NHRP Responder Field

An NHRP packet has the Responder Address extension field. This extension field is used to add the receiver's address information to this field of the reply packet when a request packet is received. This command is used to specify the address information in the Responder Address field.

	Command	Description
Step 1	Ruijie(config)#**interface tunnel** *tunnel-number*	Enters a tunnel interface to start NHRP configuration.
Step 2	Ruijie(config-Tunnel *tunnel-number*)#**ip nhrp responderinterface-name***interface-number*	Configures the first IP address of the specified interface as the address to be added to the Responder Address extension field of the NHRP reply packet.

Ruijie(config-Tunnel *tunnel-number*)#**no ip nhrp responder**	Disables the configuration.

Example 1: Add the address of the virtual-ppp1 interface to the Responder Address field on a tunnel interface.

```
Ruijie(config)#interface tunnel 1
Ruijie(config-Tunnel 1)#ip nhrp responder virtual-ppp 1
```

Configuring the Conditions for Sending NHRP Registration Request Packets

An NHC will send a registration request packet to the NHS to register its information. The following commands are used to control sending of the registration request packet.

	Command	Description
Step 1	Ruijie(config)#**interface tunnel** *tunnel-number*	Enters a tunnel interface to start NHRP configuration.

	Command	Description
Step 2	Ruijie(config-Tunnel *tunnel-number*)#**ip nhrp registration no-unique**	The **U** field is available in the NHRP registration request packet, indicating that the registration is unique. This command indicates that the registration is not unique. The value **0** of the **U** field indicates multiple registrations.
	Or:	
	Ruijie(config-Tunnel *tunnel-number*)#**ip nhrp registration timeout***seconds*	Configures the interval for sending two NHRP registration request packets. *seconds:* Indicates the interval for sending registration request packets, ranging from 1 to 65,535. The default value is 1/3 of the hold time.
	Ruijie(config-Tunnel *tunnel-number*)#**no ip nhrp registration no-unique**	Disables the configuration.
	Ruijie(config-Tunnel *tunnel-number*)#**no ip nhrp registration timeout**	Resets the default configuration.

Example 1: Configure NHRP registration requests not unique on a tunnel interface.

```
Ruijie(config)#interface tunnel1
Ruijie(config-Tunnel 1)#ip nhrp registration no-unique
```

Example 2: Configure the interval for sending NHRP registration requests as 10 seconds on the tunnel interface.

```
Ruijie(config)#interface tunnel 1
Ruijie(config-Tunnel 1)#ip nhrp registration timeout10
```

Enabling the NHRP Debug Command

Ruijie Network products provide the debug commands for monitoring and maintaining the NHRP module.

Command	Description
Ruijie#**debug nhrp**	Debugs the negotiation process of NHRP.
Ruijie#**debug nhrp cache**	Debugs cache management of NHRP.
Ruijie#**debug nhrp extension**	Debugs extension fields of NHRP.
Ruijie#**debug nhrp packet**	Debugs packet information display of NHRP.
Ruijie#**no debug nhrp**	Disables debugging by using the no command.
Ruijie#**no debug nhrp cache**	Disables debugging by using the no command.
Ruijie#**no debug nhrp extension**	Disables debugging by using the no command.
Ruijie#**no debug nhrp packet**	Disables debugging by using the no command.

Clearing NHRP Cache Information

Ruijie Network products provide the clear commands for clearing the IP-address and NBMA-address mapping table cache of the NHRP module.

Command	Description
Ruijie#**clear ip nhrp**	Clears all non-static cache information of NHRP.
Ruijie#**clear ip nhrp***ip-address*	Clears non-static NHRP cache of a specified IP-address without a mask.
Ruijie#**clear ip nhrp***ip-addressip-mask*	Clears non-static NHRP cache of a specified IP-address with a mask.

Clearing NHRP Packet Statistics

Ruijie Network products provide the clear commands for clearing NHRP packet statistics.

Command	Description
Ruijie#**clear ip nhrp counters**	Clears all NHRP packet statistics.
Ruijie#**clear ip nhrp counterstunnel***number*	Clears NHRP packet statistics of a specified tunnel interface.

Displaying

Ruijie Network products provide the show commands for displaying information of the NHRP configuration.

Command	Description
Ruijie#**show ip nhrp**	Displays all NHRP cache information.
Ruijie#**show ip nhrp brief**	Displays all NHRP cache information briefly.
Ruijie#**show ip nhrp dynamic**	Displays dynamic NHRP cache information.
Ruijie#**show ip nhrp incomplete**	Displays NHRP cache information about incomplete negotiation.
Ruijie#**show ip nhrp** *ip-address*	Displays NHRP cache information of a specified address.
Ruijie#**show ip nhrp multicast**	Displays the address for sending multicast packets.
Ruijie#**show ip nhrp nhs**	Displays NHS information.
Ruijie#**show ip nhrp static**	Displays static NHRP cache information.
Ruijie#**show ip nhrp summary**	Displays statistics about NHRP cache usage.
Ruijie#**show ip nhrp traffic**	Displays receiving and sending information of NHRP packets.
Ruijie#**show ip nhrpTunnel** *tunnel-number*	Displays NHRP cache information of a specified interface.

Example 1: Display all cache table information of NHRP. Two cache tables are available.

```
Ruijie#show ip nhrp
```

```
55.1.1.2/32 via 55.1.1.3, Tunnel 2 created 00:04:08, expire: 01:55:52
 Type: dynamic, Flags: router authoritative destination_stable unique Source_stable
 NBMA address: 11.1.1.2
 55.1.1.1/32 via 55.1.1.1, Tunnel 2 created 00:05:15, never expire
 Type: static, Flags: authoritative
 NBMA address: 11.1.1.1
```

Example 2: Display the NHRP cache table information in brief.

```
Ruijie# show ip nhrp brief
    Target      Via      NBMA    Mode   Intfc   Claimed
  55.1.1.1/32   55.1.1.1   11.1.1.1    static   Tu1     <   >
```

Example 3: Display the address information table for sending NHRP multicast packets.

```
Ruijie# show ip nhrp multicast
  I/F     NBMA address
  Tunnel2   1.1.1.1      Flags: static
  Tunnel2   2.2.2.2      Flags: static
```

Example 4: Display NHRP NHS table information.

```
Ruijie# show ip nhrp nhs
Tunnel 1  11.1.1.1   NO Responding
Tunnel 1  11.1.1.2   UP
```

Example 5: Display the NHRP cache table statistics.

```
Ruijie# show ip nhrp summary
 IP NHRP cache 0 entries, 0 bytes
 0 static  0 dynamic  0 incomplete
```

Example 6: Display the sending and receiving statistics about NHRP packets.

```
Ruijie# show ip nhrp traffic
Tunnel 1
  Sent: Total 0
        0 Resolution Request  0 Resolution Reply  0  Registration Request
        0 Registration Reply  0 Purge Request  0  Purge Reply
        0 Error Indication
  Rcvd: Total 0
        0 Resolution Request  0 Resolution Reply  0  Registration Request
        0 Registration Reply  0 Purge Request  0  Purge Reply
        0 Error Indication
```

Configuration Example

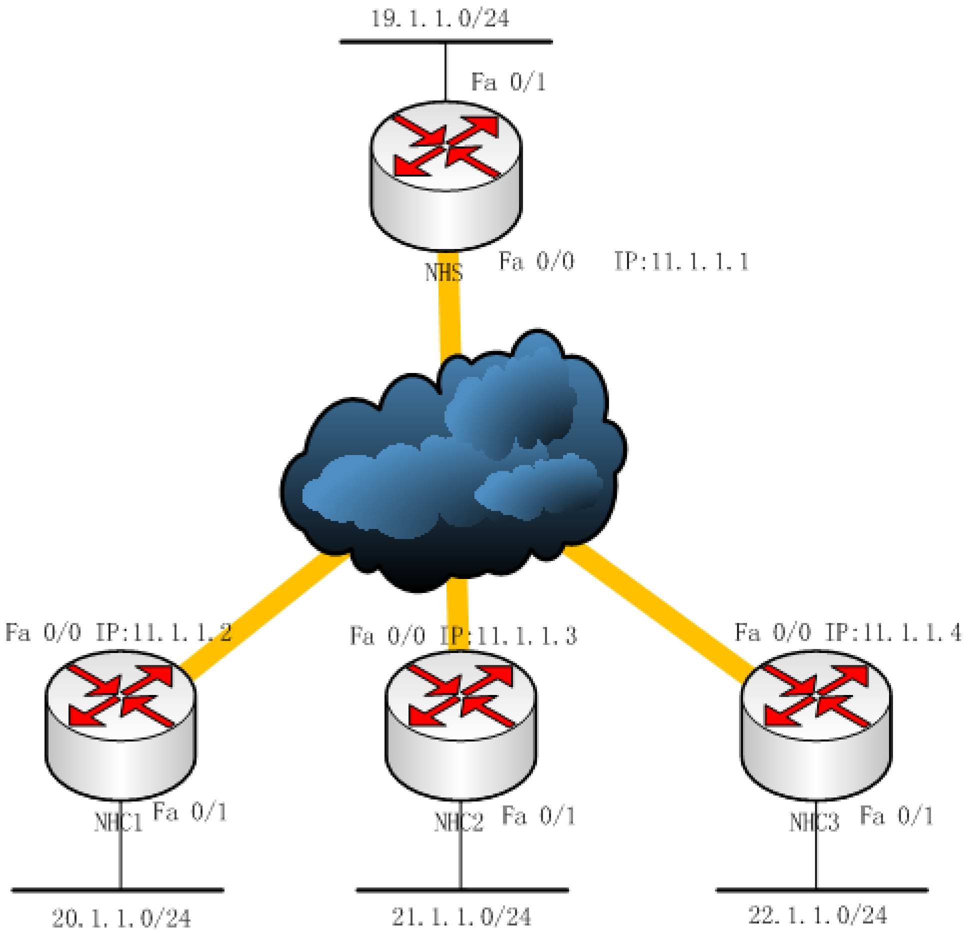
Configuring DMVPN

At present, NHRP is mainly used in a DMVPN (dynamic multi-point VPN) network. The IPSec, GRE and NHRP modules can be used together.

Networking Requirements

As shown in the following figure, four routers are needed. One router is used as the NHS and the other routers are used as the NHCs. They form a mesh topology.

Network Topology



DMVPN Network Topology

Configuration

If a mesh topology is needed, the three NHCs must be configured as MGRE tunnels. If a star topology is needed, the three NHCs must be configured as GRE tunnels.

Configuration Steps

Configuring NHS

/* Configure IPSEC. */

```
crypto isakmp policy 1
 authentication pre-share
 hash md5
!
crypto isakmp key 7 1411431e221d address 0.0.0.0 0.0.0.0
crypto ipsec transform-set trans  esp-des esp-md5-hmac
 mode transport
crypto ipsec profile vpnprof
 set transform-set trans
!
```

/* Configure the tunnel and NHRP. */

```
interface Tunnel1
ip address 55.1.1.1 255.255.255.0
no ip redirects
ip nhrp authentication test
ip nhrp map multicast dynamic
ip nhrp network-id 1000
ip nhrp holdtime 600
ip ospf network broadcast
ip ospf priority 0
ip ospf cost 1
tunnel protection ipsec profile vpnprof
tunnel source FastEthernet0/0
tunnel mode gre multipoint
!
interface FastEthernet0/0
 ip address 11.1.1.1 255.255.255.0
 duplex half
!
interface FastEthernet0/1
 ip address 19.1.1.1 255.255.255.0
 duplex half



router ospf 1
network 19.1.1.0 0.0.0.255 area 0
 network 55.1.1.0 0.0.0.255 area 0
```

Configuring NHC1

/* Configure IPSEC. */

```
crypto isakmp policy 1
 authentication pre-share
 hash md5
!
crypto isakmp key 7 1411431e221d address 0.0.0.0 0.0.0.0
crypto ipsec transform-set trans  esp-des esp-md5-hmac
 mode transport
crypto ipsec profile vpnprof
 set transform-set trans
!
```

/* Configure the tunnel and NHRP. */

```
interface Tunnel1
ip address 55.1.1.2 255.255.255.0
ip nhrp authentication test
ip nhrp map 55.1.1.1 11.1.1.1
ip nhrp map multicast 11.1.1.1
ip nhrp network-id 1000
ip nhrp holdtime 600
ip nhrp nhs 55.1.1.1
ip ospf network broadcast
ip ospf priority 0
ip ospf cost 1
tunnel protection ipsec profile vpnprof
tunnel source FastEthernet0/0
tunnel mode gre multipoint
!
interface FastEthernet0/0
 ip address 11.1.1.2 255.255.255.0
 duplex half
!
router ospf 1
network 55.1.1.0 0.0.0.255 area 0
network 20.1.1.0 0.0.0.255 area 0
```

Configuring NHC2

/* Configure IPSEC. */

```
crypto isakmp policy 1
 authentication pre-share
 hash md5
```

```
!
crypto isakmp key 7 1411431e221d address 0.0.0.0 0.0.0.0
crypto ipsec transform-set trans  esp-des esp-md5-hmac
 mode transport
crypto ipsec profile vpnprof
 set transform-set trans
!
```

/* Configure the tunnel and NHRP. */

```
interface Tunnel1
ip address 55.1.1.3 255.255.255.0
ip nhrp authentication test
ip nhrp map 55.1.1.1 11.1.1.1
ip nhrp map multicast 11.1.1.1
ip nhrp network-id 1000
ip nhrp holdtime 600
ip nhrp nhs 55.1.1.1
ip ospf network broadcast
ip ospf priority 0
ip ospf cost 1
tunnel protection ipsec profile vpnprof
tunnel source FastEthernet0/0
tunnel mode gre multipoint
!
interface FastEthernet0/0
 ip address 11.1.1.3 255.255.255.0
 duplex half
!
router ospf 1
network 55.1.1.0 0.0.0.255 area 0
network 21.1.1.0 0.0.0.255 area 0
```

Configuring NHC3

/* Configure IPSEC. */

```
crypto isakmp policy 1
 authentication pre-share
 hash md5
!
crypto isakmp key 7 1411431e221d address 0.0.0.0 0.0.0.0
crypto ipsec transform-set trans  esp-des esp-md5-hmac
 mode transport
crypto ipsec profile vpnprof
```

```
 set transform-set trans
!
```

/* Configure the tunnel and NHRP. */

```
interface Tunnel1
ip address 55.1.1.4 255.255.255.0
ip nhrp authentication test
ip nhrp map 55.1.1.1 11.1.1.1
ip nhrp map multicast 11.1.1.1
ip nhrp network-id 1000
ip nhrp holdtime 600
ip nhrp nhs 55.1.1.1
ip ospf network broadcast
ip ospf priority 0
ip ospf cost 1
tunnel protection ipsec profile vpnprof
tunnel source FastEthernet0/0
tunnel mode gre multipoint
!
interface FastEthernet0/0
 ip address 11.1.1.4 255.255.255.0
 duplex half
!
router ospf 1
network 55.1.1.0 0.0.0.255 area 0
network 22.1.1.0 0.0.0.255 area 0
```

Verification

NHS Verification

```
NHS#show ip nhrp
 55.1.1.2/32 via 55.1.1.2, Tunnel 1 created 00:01:07, expire: 01:58:53
 Type: dynamic, Flags: unique registered
 NBMA address: 11.1.1.2
55.1.1.3/32 via 55.1.1.3, Tunnel 1 created 00:01:07, expire: 01:58:53
 Type: dynamic, Flags: unique registered
 NBMA address: 11.1.1.3
55.1.1.4/32 via 55.1.1.4, Tunnel 1 created 00:01:07, expire: 01:58:53
 Type: dynamic, Flags: unique registered
 NBMA address: 11.1.1.4
```

NHC1 Verification

```
NHS#show ip nhrp
55.1.1.1/32 via 55.1.1.1, Tunnel 1 created 00:02:14, never expire
 Type: static, Flags: authoritative
 NBMA address: 11.1.1.1
55.1.1.3/32 via 55.1.1.3, Tunnel 1 created 00:20:46, expire: 01:39:14
 Type: dynamic, Flags: router authoritative destination_stable unique Source_stable
 NBMA address: 11.1.1.3
55.1.1.4/32 via 55.1.1.4, Tunnel 1 created 00:20:46, expire: 01:39:14
 Type: dynamic, Flags: router authoritative destination_stable unique Source_stable
 NBMA address: 11.1.1.4
```

NHC2 Verification

```
NHS#show ip nhrp
55.1.1.1/32 via 55.1.1.1, Tunnel 1 created 00:02:14, never expire
 Type: static, Flags: authoritative
 NBMA address: 11.1.1.1
55.1.1.2/32 via 55.1.1.2, Tunnel 1 created 00:20:46, expire: 01:39:14
 Type: dynamic, Flags: router authoritative destination_stable unique Source_stable
 NBMA address: 11.1.1.3
55.1.1.4/32 via 55.1.1.4, Tunnel 1 created 00:20:46, expire: 01:39:14
 Type: dynamic, Flags: router authoritative destination_stable unique Source_stable
 NBMA address: 11.1.1.4
```

NHC3 Verification

```
NHS#show ip nhrp
55.1.1.1/32 via 55.1.1.1, Tunnel 1 created 00:02:14, never expire
 Type: static, Flags: authoritative
 NBMA address: 11.1.1.1
55.1.1.3/32 via 55.1.1.3, Tunnel 1 created 00:00:30, expire: 01:59:30
 Type: dynamic, Flags: router authoritative destination_stable unique Source_stable
 NBMA address: 11.1.1.3
55.1.1.2/32 via 55.1.1.2, Tunnel 1 created 00:00:30, expire: 01:59:30
 Type: dynamic, Flags: router authoritative destination_stable unique Source_stable
 NBMA address: 11.1.1.2
```

Security Configuration

18. Configuring RPL

Configuring ACLs

Overview

As part of Ruijie's security solution, an access control list (ACL) is used to provide a powerful traffic filtering function. Currently, Ruijie products support the following ACLs:

- Standard and extended IP ACLs
- IPv6 Extended ACLs

Depending on networks conditions, you can choose different ACLs to control data flows.

ACL Introduction

An ACL is also referred to as a firewall or packet filtering. ACLs permit or discard packets on interfaces of network devices by defining rules. According to application scopes, they can be divided into ACLs and QoS ACLs.

By filtering the data streams, you can restrict the communication data types in the network and restrict the users of the network and the device they can use. When data streams pass the switch, ACLs classify and filter them, that is, check the data streams input from the specified interface and determine whether to permit or deny them according to the matching conditions.

To sum up, the security ACL is used to control which data flow is allowed to pass through the network device. The QoS policy performs priority classification and processing for the data flow.

ACLs consist of a series of entries, known as Access Control Entry (ACE). Each entry has its matching condition and behavior.

ACL rules can be applied to the source addresses, destination addresses, upper layer protocols, time ranges or other information of data flows.

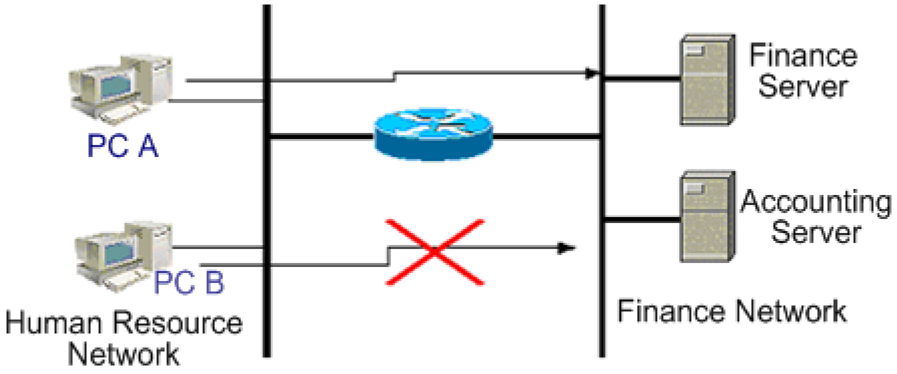
Why to Configure ACLs

There are many reasons why ACLs need to be configured. In most cases, ACLs are used to:

- Restrict route update: Control where the route update information is sent and received.
- Restrict network access: To ensure network security, provide users with access to desired services only (for example, if a user only needs webpage access and email services, other services such as Telnet are disabled), specify a time period in which access is permitted, or specify hosts which are allowed to access Internet.

In Figure 1, only host A is allowed to access Finance Network, while Host B is not.

Figure 1 Using ACLs to control network access



When to Configure Access Lists

Depending on your requirements, you can select the basic ACL or dynamic ACL. In general, the basic ACL can meet the security requirement. However, experienced hackers may use some software to forge source addresses and spoof the devices so as to gain access. Before the user can access the network, the dynamic ACL requires authentication so that the hackers are difficult to invade the network. So, in some sensitive areas the dynamic ACL can be used to ensure the network security.

Note

An inherent problem of all ACLs is spoofing, the behavior of providing spoofed source addresses to deceive switches. This cannot be avoided even you use the dynamic ACL. During the effective access period of an authenticated user, a hacker may use a spoofed user address and accesses the network. There are two methods to resolve the problem. One method is to set free time for a user to access the network as little as possible, making it hard for a hacker to attack the network. The other method is to use IPSEC to encrypt network data, ensuring that all the data entering switches is encrypted.

ACLs are usually configured in the following positions of network devices:

- Devices between the internal network and external network (such as the Internet)
- Devices at the border of two parts in a network
- Devices on the access control port

The execution of the ACL statements must follow the order in the table strictly. Starting from the first statement, once the header of a packet matches a conditional judge statement in the table, the following statements are ignored.

Input/Output ACL, Filtering Domain Template and Rule

When a device interface receives a message, the input ACL checks whether the message matches an ACE of the ACL input on the interface. When a device interface is ready to output a message, the output ACL checks whether the message matches an ACE of the ACL output on the interface.

When detailed filtering rules are formulated, all or some of the preceding eight items may be used. As long as the message matches one ACE, the ACL processes the message as the ACE defined (permit or deny). The ACE of an ACL identifies Ethernet messages according to some fields of Ethernet messages. The fields include the following:

Layer-2 fields:

- 48-bit source MAC address (all the 48 bits must be declared)
- 48-bit destination MAC address (all the 48 bits must be declared)
- 16-bit layer-2 type field

Layer 3 fields:

- Source IP address field (you can specify all the 32 bits of the IP address, or specify a type of streams of the defined subnet)
- Destination IP address field (you can specify all the 32 bits of the IP address, or specify a type of streams of the defined subnet)
- Protocol type fields

Layer-4 fields:

- You can specify one UDP source port, destination port, or both
- You can specify one UDP source port, destination port, or both

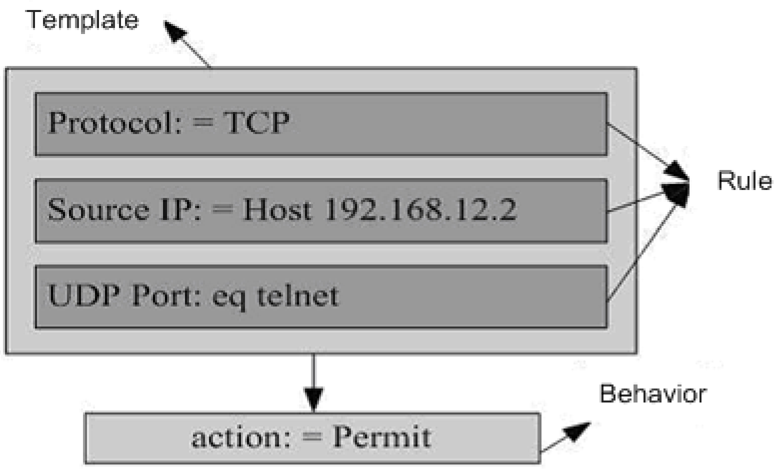
The filtering domain consists of the fields in the packets based on which the packets are identified and classified when you create an ACE. A filtering domain template is the definition formed by these fields. For example, when one ACE is generated, you want to identify and classify messages according to the destination IP field of a message. When another ACE is generated, you want to identify and classify messages according to the source IP address field of a message and the source port field of UDP. In this way, these two ACEs use different filtering domain templates.

Rules refer to the values of the ACE mask. For example, one ACE is:

```
permit tcp host 192.168.12.2 any eq telnet
```

In this ACE, the filtering domain template is a collection of the following fields: Source IP Address Fields, IP Protocol Fields and Destination TCP Port Fields. Corresponding values (rules) are respectively as follows: Source IP Address=Host 192.168.12.2; IP Protocol=TCP; TCP Destination Port=Telnet.

Figure 2 Analysis of the ACE: permit tcp host 192.168.12.2 any eq telnet



A filtering domain template can be the collection of L3 fields and L4 fields or the collection of multiple L2 fields. However, the filtering domain templates of a standard and extended ACL cannot be the collection of L2 and L3, L2 and 4, L2 and L3, or L4 fields.

Note

Caution

1. When associating SVI with the ACL at the outbound direction, you should note that:

Standard IP ACL and extended IP ACL are supported. There are some limits on matching the destination IP address and the destination MAC address with an ACL. If you need to match the destination IP address not in the subnet IP range of the associated SVI in the standard IP ACL and extended IP ACL, this ACL will not take effect. For example, the IP address of VLAN 1 is 192.168.64.1 and subnet mask of VLAN 1 is 255.255.255.0. Now you create an ACL with the ACE of **deny udp any 192.168.65.1 0.0.0.255 eq 255** and apply this ACL to the egress of VLAN 1. This ACL will not function for the destination IP address is not in the subnet IP range of VLAN 1. If the ACE is **deny udp any 192.168.64.1 0.0.0.255 eq 255**, this ACL will take effect.

Note

When applying an ACL, the labeled MPLS packet matching does not take effect if an ACE in the ACL (including the IP ACL) matches a non-L2 field (such as SIP and DIP).

Configuring IP Access List

To configure ACLs on a device, you must specify unique names or numbers for the ACLs of a protocol to uniquely identify each ACL within the protocol. The following table lists the protocols that can use numbers to specify ACLs and the number ranges of ACLs that can be used by each protocol.

Protocol	Number Range
Standard IP	1-99, 1300 - 1999
Extended IP	100-199, 2000 - 2699

Guide to Configuring IP ACLs

When you create an ACL, defined rules will be applied to all packets on a device. The device decides whether to forward or block a packet by judging whether the packet matches a rule.

Basic ACLs are classified into standard ACLs and extended ACLs. The typical rules defined in ACLs are as follows:

- Source address
- Destination address
- Upper layer protocol
- Time range

Standard IP ACLs (numbered from 1 to 99 and from 1300 to 1999) forward or block packets according to source addresses. Extended IP ACLs (numbered from 100 to 199 and from 2000 to 2699) use the above four combinations to forward or block packets. Other types of ACLs forward or block packets according to related codes.

A single ACL can use multiple separate ACL statements to define multiple rules. Where, all statements use the same number or name to bind these statements to the same ACL. However, the more the used statements are, the more difficult to read and understand an ACL.

Statement containing an implicit deny statement for all packets

The end of the ACL contains an implicit deny statement for all packets that it did not find a match for before reaching the end, as shown in the following example:

```
access-list 1 permit host 192.168.4.12
```

This ACL allows only the messages destined for host 192.168.4.12. This is because the ACL contains the following rule statement at the end: **access-list 1 deny any**

Here is another example:

```
access-list 1 deny host 192.168.4.12
```

If the ACL contains the only preceding statement, the messages from any host will be denied on the port.

Note

It is required to consider the routing update message when defining the ACL. Since the end of the ACL contains an implicit deny statement for all packets, this may cause all routing update messages blocked.

☑ If the inserted line cards do not include EA series, the ACEL associated with the outgoing direction of the AP port has no default deny ACE, which shall be configured manually as needed.

Order to Input Rule Sentences

Each added rule is appended to the ACL. If a statement is created, then you cannot delete it separately but delete the whole ACL. Therefore, the order of ACL statements is very important. When deciding whether to forward or block packets, a device compares packets and statements in order of statement creation time until It finds a matching statement.

If you have created a statement that allows all packets to pass, then the following statements will not be checked, as shown in the following example:

```
access-list 101 deny ip any any
access-list 101 permit tcp 192.168.12.0 0.0.0.255  eq  telnet any
```

Because the first rule statement denies all IP packets, the packets for accessing host 192.168.12.0/24 will be denied. Because the device discovers that the packets match the first rule statement, it will not check other rule statements.

Configuring IP ACLs

The configuration of the basic ACL includes the following steps:

- Define a basic ACL
- Apply the ACL to a specific interface.

There are two methods to configure a basic ACL.

Method 1: Run the following command in global configuration mode:

Command	Function
Ruijie(config)# **access-list id** {**deny** | **permit**} {**src** *src-wildcard* | **host src** | **any** | *interface idx*} [**time-range** *tm-rng-name*]	Defines an ACL .
Ruijie(config)# **interface** *interface*	Selects the interface to which the ACL is to be applied.
Ruijie(config-if)# ip access-group id { in | out } [unreflect]	Applies the ACL to the specific interface

Method 2: Run the following command in ACL configuration mode:

Command	Function
Ruijie(config)# **ip access-list** { **standard** | **extended** } { *id* | *name* }	Enters ACL configuration mode.
Ruijie (config-xxx-nacl)# [**sn**] { **permit** | **deny** } {**src** *src-wildcard* | **host src** | **any** } [**time-range** *tm-rng-name*]	Adds ACEs to the ACL. For details about the command, see command reference.
Ruijie(config-xxx-nacl)# **exit**	Exits ACL mode.
Ruijie(config)# **interface** *interface*	Selects the interface to which the ACL is to be applied.
Ruijie(config-if)# ip access-group *id* { in | out } [unreflect]	Applies the ACL to the specific interface.

Note

In method 1, an ACL can only be numbered. In method 2, an ACL can be numbered and named, and ACE priority can be specified if available. By default, the reflexive ACL is enabled on the IP ACL port. You can run the **unreflect** command to disable the reflexive ACL. (The operation principles of the reflexive ACL are described as follows:

a. The router automatically generates a temporary ACL according to the L3 and L4 information of the originating traffic in the internal network based on the principles. That is, the protocol is constant, while the source and destination IP addresses, and the source and destination ports are rigidly exchanged.

b. The router allows traffic to enter the internal network only when the L3 and L4 information of the returned traffic strictly matches the information in the temporary ACL previously created based on the outputting traffic.)

Displaying IP ACLs

To monitor ACLs, run the following command in privileged user mode:

Command	Function
Ruijie# **show access-lists** [*id* | *name*]	Queries the basic ACLs.

Configuring IPv6 Extended ACLs

Configuring IPv6 Extended ACLs

The configuration of an IPv6 ACL includes the following steps:

- Define an IPv6 ACL

- Apply the ACL to a specific interface (application particular case)

To configure a basic ACL, run the following command in ACL configuration mode:

Command	**Function**
Ruijie(config)# **ipv6 access-list** *name*	Enters ACL configuration mode.
Ruijie (config-ipv6-nacl)# [sn] **{permit | deny** } prot {*src-ipv6-prefix/prefix-len* | **host** *src-ipv6-addr* | **any**} *{dst-ipv6-pfix/pfix-len* | **any | host** *dst-ipv6-addr*} [**dscp** *dscp*] [**flow-label** *flow-labe*l] [**time-range** *tm-rng-name*]	Adds ACEs to the ACL. For details about the command, see command reference.
Ruijie(config-exp-nacl)# **exit**	Exits the access control list mode.
Ruijie(config)# **interface** *interface*	Selects the interface to which the ACL is to be applied.
Ruijie(config-if)# **ipv6 traffic-filter** *name* {**in** | **out**}	Applies the ACL to the specific interface.

Displaying Configuration of IPv6 Extended ACLs

To monitor ACLs, run the following command in privileged user mode:

Command	**Function**
show access-lists [*name*]	Queries the basic ACLs.

Configuring Extended Expert ACLs

To configure Expert extended ACLs on a device, you must specify unique names or numbers for the ACLs of a protocol to uniquely identify each ACL within the protocol. The following table lists the number range of the Expert ACLs.

Protocol	**Number Range**
Extended Expert ACL	2700-2899

Guide to Configuring Expert Extended ACLs

When you create an expert extended ACL, defined rules will be applied to all packets on a device. The device decides whether to forward or block a packet by judging whether the packet matches a rule.

The typical rules defined in Expert ACLs are as follows:

- All information in basic ACLs and MAC extended ACLs
- VLAN ID

Extended Expert ACLs (numbered from 2700 to 2899) are the syntheses of basic ACLs and MAC extended ACLs and can filter VLAN IDs.

A single expert ACL can use multiple separate ACL statements to define multiple rules. Where, all statements use the same number or name to bind these statements to the same ACL.

Configuring an Expert Extended ACL

The configuration of an expert ACL includes the following steps:

- Define an expert ACL
- Apply the ACL to a specific interface (application particular case)

There are two methods to configure an Expert ACL.

Method 1: Run the following command in global configuration mode:

Command	**Function**
Ruijie (config)# **access-list** *id* {**deny** | **permit**} [*prot* | {[*ethernet-type*] [**cos** *cos*]}] [**VID** *vid*] {**src** *src-wildcard* | **host** *src* | **interface** *idx*} {**host** *src-mac-addr* | **any**} {**dst** *dst-wildcard* | **host** *dst* | **any**}{**host** *dst-mac-addr* | **any**}] [**precedence** *precedence*] [**tos** *tos*] [**dscp** *dscp*] [**fragment**] [**time-range** *tm-rng-name*]	Defines an ACL. For details about the command, see command reference.
Ruijie(config)# **interface** *interface*	Selects the interface to which the ACL is to be applied.
Ruijie(config-if)# expert access-group *id* {in | out } [unreflect]	Applies the ACL to the specific interface.

Method 2: Run the following command in ACL configuration mode:

Command	**Function**
Ruijie(config)# **expert access-list extended** {*id*|*name*}	Enters ACL configuration mode.
Ruijie (config-exp-nacl)# [*sn*]{ **permit** | **deny** }[*prot* | {[*ethernet-type*] [**cos** *cos*]}] [**VID** *vid*] {**src** *src-wildcard* | **host** *src* | **interface** *idx*}{**host** *src-mac-addr* | **any**} {**dst** *dst-wildcard* | **host** *dst* | **any**} {**host** *dst-mac-addr* | **any**}][**precedence** *precedence*] [**tos** *tos*] [**dscp** *dscp*] [**fragment**] [**time-range** *tm-rng-name*]	Adds ACEs to the ACL. For details about the command, see command reference.
Ruijie(config-exp-nacl)# **exit**	Exit ACL mode.
Ruijie(config)# **interface** *interface*	Selects the interface to which the ACL is to be applied.
Ruijie(config-if)# **expert access-group** {*id*|*name*} {**in**|**out**} **[unreflect]**	Applies the ACL to the specific interface.

Note

In method 1, an ACL can only be numbered. In method 2, an ACL can be numbered and named, and ACE priority can be specified if available. In a version supporting ACE priority, method 2 can also specify the priorities of ACEs (using the [*sn*] option in a command).

☑ The router supports neither packet fragment filtering nor Expert ACLs.

Displaying Configuration of Expert Extended ACLs

To monitor ACLs, run the following command in privileged user mode:

Command	**Function**
show access-lists [*id* | *name*]	Queries the Expert ACLs.

Configuring MAC Extended ACLs

To configure MAC extended ACLs on a device, you must specify unique names or numbers for the ACLs of a protocol to uniquely identify each ACL within the protocol. The following table lists the range of the numbers that can be used to specify MAC ACLs.

Protocol	**Number Range**
MAC Extended Access List	700-799

Guide to Configuring MAC Extended ACLs

When a MAC ACL is created, the defined rules will be applied to all packets on a device. The device decides whether to forward or block a packet by judging whether the packet matches a rule.

The typical rules defined in MAC ACLs are as follows:

- Source MAC address
- Destination MAC address
- Ethernet protocol type
- Time-range

The MAC extended ACLs (numbered from 700 to 799) forward or block the packets based on the source and destination MAC addresses, and can also match Ethernet packets.

A single MAC ACL can use multiple separate ACL statements to define multiple rules. Where, all statements use the same number or name to bind these statements to the same ACL.

Configuring a MAC Extended ACL

The configuration of an MAC ACL includes the following steps:

- Define an MAC ACL
- Apply the ACL to a specific interface

There are two methods to configure an MAC ACL.

Method 1: Run the following command in global configuration mode:

Command	**Function**
Ruijie(config)# **access-list** id {**deny** | **permit**}{**any** | **host** *src-mac-addr*} {**any** | **host** dst-mac-addr} [*ethernet-type*] [**cos** *cos*]	Defines an ACL. For details about the command, see command reference.
Ruijie(config)# **interface** *interface*	Selects the interface to which the ACL is to be applied.
Ruijie(config-if)# **mac access-group** *id* { **in** | **out** }	Applies the ACL to the specific interface.

Method 2: Run the following command in ACL configuration mode:

Command	**Function**
Ruijie(config)# **mac access-list extended** {*id* | *name*}	Enters ACL configuration mode.

Ruijie (config-mac-nacl)# [sn] { **permit** | **deny** }{**any** | **host** *src-mac-addr*} {**any** | **host** *dst-mac-addr*} [*ethernet-type*] [**cos** *cos*]	Adds ACEs to the ACL. For details about the command, see command reference.
Ruijie(config-mac-nacl)# **exit** Ruijie(config)# **interface** *interface*	Exits ACL mode and selects the interface to which the ACL is to be applied.
Ruijie(config-if)# **mac access-group** {*id* | *name*} { **in** | **out**}	Applies the ACL to the specific interface.

Note

Method 1 only configures the numerical value ACL. Method 2 can configure the names and numerical value ACL, and specify the table entry priorities (in the devices that support ACE priorities).

☑ The route does not support MAC ACLs.

Displaying Configuration of MAC Extended ACLs

To monitor ACLs, run the following command in privileged EXEC mode:

Command	Function
show access-lists [*id* | *name*]	Queries the basic ACLs.

Other Related Configurations

Configuring ACEs by Priority

To embody the ACE priority, criteria is set up for each ACL so that ACEs in an ACL are arranged in a standard manner: using an ACE sequence number as the start point and making the sequence number grows at an increment:

- ACEs are arranged by sequence number in ascending order in the chain table.
- ACE arrangement starts from a sequence number. If no number is specified, it increases at an increment on the basis of the previous ACE number.
- To specify the sequence number of an ACE, insert the ACE and ensure that a new ACE can be inserted between two adjacent ACEs.
- The ACL specifies the start number and the increment.

The **ip access-list resequence** {*acl-id*| *acl-name*} *sn-start sn-inc* command is available. For details, see command reference.

Whenever the preceding command is run, the ACEs in the ACL will be sorted. For example, the ACEs in the ACL named **tst_acl** are numbered as follows:

In the beginning

```
ace1: 10
ace2: 20
ace3: 30
```

The ACEs are numbered as follows after "the **ip access-list resequence** *tst_acl 100 3*" command is run:

```
Ruijie(config)# ip access-list resequence tst_acl 100 3
ace1: 100
ace2: 103
ace3: 106
```

If you do not specify *sn-num* when adding ACE 4, ACE 4 is numbered as follows:

```
Ruijie(config-std-nacl)# permit …
ace1: 100
ace2: 103
ace3: 106
ace4: 109
```

If you set *seg-num* to **105** when adding ACE 5, ACE 5 is numbered as follows:

```
Ruijie(config-std-nacl)# 105 permit …
ace1: 100
ace2: 103
ace5: 105
ace3: 106
ace4: 109
```

The sequence number mechanism is designed to add ACEs by priority.

Delete ACEs

```
Ruijie(config-std-nacl)# no 106
ace1: 100
ace2: 103
ace5: 105
ace4: 109
```

It is also convenient to delete ACEs with a sequence number.

Configuring ACL Logging

When ACL Logging is enabled, if a packet matches a logging-enabled ACE and the matching speed reaches or even exceeds the configured logging threshold, the system generates a log within one logging interval to determine whether to permit or deny this packet.

Note

This function applies only to standard and extended IP ACLs and is optional.

Default configuration

The ACL logging function is disabled by default.

ACL Options are configured as follows:

Configure the ACL logging speed threshold. This threshold means the maximum speed an ACE is matched. When it is exceeded, a log is generated.

Command	Function
Ruijie(config)#**ip access-list log-update threshold** *threshold-value*	Configures the ACL logging threshold.

Configure the ACL logging interval, in milliseconds.

Command	Function
Ruijie(config)#**ip ccess-list logging interval** *interval-value*	Configures the ACL logging interval.

Enable ACE logging so that packets matching an ACE can be counted.

Command	Function
Ruijie(config)# **ip access-list extended** { *id* | *name* }	Enters ACL configuration mode.
Ruijie(config-ext-nacl)# [sn] {**permit** | **deny**} **protocol** *source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** *tos*] [**fragment**] [**range** *lower upper*] [**time-range** *time-range-name*] [**option** *option*] [**log**]	Adds ACEs to the ACL. For details about the command, see command reference.
Ruijie(config-exp-nacl)# **exit**	Exits ACL configuration mode and selects the interface to which the ACL is to be applied.

Configuration example:

- Configure the permission and password for enabling the ACL logging function.

```
Ruijie> enable
Ruijie#
```

- Enter global configuration mode.

```
Ruijie# configure terminal
Ruijie(config)#
```

- Configure the ACL logging threshold and interval.

```
Ruijie(config)# ip access-list log-update threshold 1
Ruijie(config)# ip access-list logging interval 1
```

- Enter ACL configuration mode and enable logging on the desired ACE.

```
Ruijie(config-ext-nacl)# permit ip 99.9.9.0 0.0.0.255 any log
```

- Add a deny ACE and enable the logging function on the ACE.

```
Ruijie(config-ext-nacl)# deny ip any any log
```

- end

```
Ruijie(config-ext-nacl)# end
```

- The following log will be generated when the ACL logging threshold is reached or exceeded:

```
*Feb 20 14:10:48.747: %SEC-6-IPACCESSLOGNP: list s1 permitted 0 99.9.9.2 -> 99.9.9.1, 1 packet
```

```
*Feb 20 14:11:37.171: %SEC-6-IPACCESSLOGNP: list s1 permitted 0 99.9.9.2 -> 99.9.9.1, 2 packets
*Feb 20 14:42:51.207: %SEC-6-IPACCESSLOGNP: list s1 denied 0 90.9.9.2 -> 99.9.9.1, 1 packet
```

In privileged configuration mode, run the following commands to configure a global security tunnel:

Command	Function
Ruijie# configure terminal	Enters global configuration mode.
Ruijie(config)#security global access-group *1*	Configures a global security tunnel.

In privileged configuration mode, execute the following commands to set an exception port:

Command	Function
Ruijie# configure terminal	Enters global configuration mode.
Ruijie(config)#**interface** *idx*	Enters interface configuration mode.
Ruijie(config-if)# security uplink enable	Sets the interface as an exceptional port..

In privileged configuration mode, run the following commands to configure a security tunnel on the interface:

Command	Function
Ruijie# configure terminal	Enters global configuration mode.
Ruijie(config)#**interface** *idx*	Enters interface configuration mode.
Ruijie(config-if)# **security access-group** *1*	Configures a security tunnel on the interface.

The following example shows how to configure a security tunnel on a security port where IP/MAC binding is configured, so that IPX packets can pass:

Set port 4 as the security port and bind IP address and MAC address

```
Ruijie(config)#interface FastEthernet 0/4
Ruijie(config-if)#switchport port-security
Ruijie(config-if)#switchport port-security binding 0000.0000.0011 vlan 1 192.168.6.3
```

Only the packets whose source IP address is 192.168.6.3 and MAC address is 0000.0000.0011 can pass the device through port 4. To receive IPX packets, set a security tunnel as follows:

```
Ruijie#configure
Ruijie(config)#expert access-list extended safe_channel
Ruijie(config-exp-nacl)#permit ipx any any
Ruijie(config-exp-nacl)#exit
Ruijie(config)#security global access-group safe_channel
```

Or configure a security tunnel on the interface:

```
Ruijie#configure
Ruijie(config)#expert access-list extended safe_channel
Ruijie(config-exp-nacl)#permit ipx any any
Ruijie(config-exp-nacl)#exit
Ruijie(config)#interface FastEthernet 0/4
```

Ruijie(config-if)#**security access-group** *safe_channel*

IPX packets can pass through port 4 after a security channel is configured globally or on an interface.

Configuring ACL80

ACL 80 is also called the custom ACL, which is used to match the first 80 bytes of a packet. A packet consists of a series of byte flows. ACL 80 enables the user to filter packets according to the specified 16 bytes of the first 80 bytes in the packet.

Note

The SMAC/DMAC/SIP/DIP/ETYPE field of the packets is not specified. In other words, a packet is filtered only when the specified 16 bytes match ACL 80 in addition to these fields.

For any 16-byte field, it is possible to compare or not the configured value by bits. In other words, it allows setting any bit of those 16 bytes as 0 or 1. There are two factors in filtering any byte: filtering rule and filter domain template. The bits of the both are one-to-one corresponding. The filtering rule specifies the value of the field to be filtered. The filter domain template specifies whether to filter the related fields in the filtering rule (“1” indicates matching the bit in the corresponding filtering rule, 0 for not). Therefore, when it is time to match a bit, it is required to set 1 for the corresponding bit in the filter domain template. If the filter domain template bit is set as 0, no match will be done no matter what the corresponding bit is in the filtering rule.

For example,

```
Ruijie(config)# expert access-list advanced name
Ruijie(config-exp-dacl)# permit 00d0f8123456 ffffffffffff 0
Ruijie(config-exp-dacl)# deny 00d0f8654321 ffffffffffff 6
```

The user custom ACL matches any byte of the first 80 bytes in the layer-2 data frames according to the user definitions, and then performs corresponding processing for the packets. To use ACL 80 correctly, it is necessary to have in-depth knowledge about the structure of layer-2 data frames. The following illustrates the first 64 bytes in a layer-2 data frame (each letter indicates a hexadecimal number, and each two letters indicate a byte).

AA AA AA AA AA AA BB BB BB BB BB BB CC CC DD DD

DD DD EE FF GG HH HH HH II II JJ KK LL LL MM MM

NN NN OO PP QQ QQ RR RR RR RR SS SS SS SS TT TT

UU UU VV VV VV VV WW WW WW WW XY ZZ aa aa bb bb

The following table lists the meanings and offset values of each letter:

Letter	Meaning	Offset	Letter	Meaning	Offset
A	Destination MAC	0	O	TTL field	34
B	Source MAC	6	P	Protocol ID	35
C	VLAN tag field	12	Q	IP checksum	36
D	Data frame length field	14	R	Source IP address	38
E	DSAP field	18	S	Destination IP address	42
F	SSAP field	19	T	TCP source port	46
G	Ctrl field	20	U	TCP destination port	48
H	Org Code field	21	V	Sequence number	50
I	Encapsulated data type	24	W	Confirmation field	54

J	IP version No.	26	XY	IP header length and reservation bits	58
K	TOS field	27	Z	Reservation bit and flags bit	59
L	IP packet length	28	a	Windows size field	60
M	ID	30	b	Others	62
N	Flags field	32			

As shown in the preceding table, the offset of each field is the offset in the SNAP+tag 802.3 data frame. In ACL 80, the user can use two parameters, the rule mask and offset, to abstract any byte from the first 80 bytes of the data frame, and then compare it with the user defined rule to filter the matched data frame for corresponding processing. The user defined rule can be some fixed attributes of the data. For example, the user wants to filter all the TCP packets by defining the rule as “06”, rule mask as “FF” and offset as 35. Here, the rule mask and offset work together to abstract the contents of the TCP protocol ID field in the received data frame, and compare it with the rule to filter all TCP packets.

Caution

ACL 80 can be used to match Ethernet packets, 803.3 SNAP packets, and 802.311c packets. If the value for matching DSAP to the cnt1 field is set to AAAA03, it indicates to match the 803.3 SNAP packets. If the value is set to E0E003, it indicates to match the 803.311c packets. This field cannot be set to match Ethernet packets.

Caution

ACL 80 only match only the 16 bytes of a packet. If the 16 bytes are used, no fields other than the 16 bytes can be matched. For example:

```
Ruijie(config)# expert access-list advanced name
Ruijie(config-exp-dacl)# permit 11223344556677889900aabbccd deeff
ffffffffffffffffffffffffffffffff 50
```

Add another ACE:

```
Ruijie(config-exp-dacl)#permit 11223344556677889900aabbccd deeff
ffffffffffffffffffffffffffffffff 54
```

The configuration will fail because the 16 bytes are used by the first ACE. To match the second ACE, you must delete the first ACE.

Configuring IP Options Filtering

IP Options filtering is used to match options in the IP packet header by option value (0–255) or option name. If the IP options in a packet match all bits defined in ACEs, the packet is deemed to match the ACL. Users can set any values for IP options to filter packets with specified IP options.

Note

This feature applies only to named extended ACLs and is optional.

Configure IP Options filtering:

Command	Function
Ruijie(config)# **ip access-list extended** { *id* | *name* }	Enters ACL configuration mode.
Ruijie(config-ext-nacl)# [sn] {**permit** | **deny**} **protocol** *source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** *tos*] [**fragment**] [**range** *lower upper*] [**time-range** *time-range-name*] [**option** *option*] [**log**]	Adds ACEs to the ACL. For details about the command, see command reference.
Ruijie(config-exp-nacl)# **exit**	Exits ACL configuration mode and selects the interface to which the ACL is to be applied.
Or	
Ruijie(config)# **interface** *interface*	Exits ACL configuration mode and selects the interface to which the ACL is to be applied.
Ruijie(config-if)# **ip access-group** {*id* | *name*} {**in** | **out**}	Applies the ACL to the specific interface.

Configuration example:

- Configure the permission and password for enabling the IP Option feature.

```
Ruijie> enable
Ruijie#
```

- Enter global configuration mode.

```
Ruijie# configure terminal
Ruijie(config)#
```

- Enter ACL configuration mode.

```
Ruijie(config)# ip access-list extended ip-options
Ruijie(config-ext-nacl)#
```

- Add ACEs.

```
Ruijie(config-ext-nacl)# permit ip any any option lsr
```

- Add deny ACEs.

```
Ruijie(config-ext-nacl)# deny ip any any option any-options
```

- end

```
Ruijie(config-ext-nacl)# end
```

- Display the configuration result.

```
Ruijie# show  access-list ip-options
ip access-lists extended ip-options
10 permit tcp any any option lsr
20 deny tcp any any option any-options
```

Configuring ACLs Based on the Time Range

You can make ACLs effective based on time, for example, ACLs take effect during certain periods in a week. For this purpose, you must first set a time range.

Time range implementation depends on the system clock. If you want to use this function, you must assure that the system has a reliable clock.

In privileged configuration mode, run the following commands:

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **time-range** *time-range-name*	Identifies a time range by using a meaningful display character string as its name
Ruijie(config-time-range)# **absolute** [**start time** *date*] **end time** *date*	Sets the absolute time range (optional). For details, see the time range configuration guide.
Ruijie(config-time-range)# **periodic day-of-the-week time to** [*day-of-the-week*] **time**	Sets the periodic time range (optional).
Ruijie# **show time-range**	Verifies the configuration.
Ruijie# **copy running-config startup-config**	Saves the configuration.
Ruijie(config)# **ip access-list extended** *101*	Enters ACL configuration mode.
Ruijie(config-ext-nacl)# **permit ip any any time-range** *time-range-name*	Configures a time range-based ACE.

Note

The length of the name should be 1-32 characters without any blank space.

You can set one absolute time range at most. The application based on time-ranges will be effective only in this time range.

You can set one or more intervals. If you have already set a running time range for **time-range**, the application takes effect at intervals in that time range.

The following example shows how to deny HTTP packets during the working hours in a week by using the time range-based ACLs:

```
Ruijie(config)# time-range no-http
Ruijie(config-time-range)# periodic weekdays 8:00 to 18:00
Ruijie(config)# end
Ruijie(config)# ip access-list extended limit-udp
Ruijie(config-ext-nacl)# deny tcp any any eq www time-range no-http
Ruijie(config-ext-nacl)# exit
Ruijie(config)# interface gigabitEthernet 0/1
Ruijie(config-if)# ip access-group no-http in
Ruijie(config)# end
```

Example of time ranges:

```
Ruijie# show time-range
time-range entry: no-http(inactive)
```

```
periodic Weekdays 8:00 to 18:00
time-range entry: no-udp
periodic Tuesday 15:30 to 16:30
```

Configuring TCP Flag Filtering

The TCP Flag filtering feature provides a flexible mechanism. At present, TCP Flag filtering control supports the match-all option. Namely, when the TCP Flags in a received packet exactly match those defined in the ACE, the packet will be checked by the ACL rule. A user can define any combination of TCP Flags to filter some packets with specific TCP Flags.

For example,

```
permit tcp any any match-all rst
```

Allow the packets to pass if the TCP Flag is reset and other fields are set to 0.

Note

This feature is optional when the protocol number is set to TCP in naming ACLs and numerical value ACLs. MAC extended and IP standard ACLs do not support this function.

To configure TCP Flag filtering, run the following commands:

Command	Function
Ruijie(config)# **ip access-list extended** { *id* | *name* }	Enters ACL configuration mode
Ruijie(config-ext-nacl)# [sn] {**permit** | **deny**} **tcp** source source-wildcard [operator port] destination destination-wildcard [operator port] [**match-all** flag-name][**precedence** precedence]	Adds ACEs to the ACL. For details about the command, see command reference.
Ruijie(config-exp-nacl)# **exit**	Exits ACL configuration mode and selects the interface to which the ACL is to be applied.
Or	
Ruijie(config)# **interface** *interface*	Exits ACL configuration mode and selects the interface to which the ACL is to be applied.
Ruijie(config-if)# **ip access-group** {*id* | *name*} {**in** | **out**}	Applies the ACL to the specific interface

The following example explains how to configure TCP Flag filtering.

- Configure the permission and password for enabling the TCP Flag filtering function.

```
Ruijie> enable
Ruijie#
```

- Enter global configuration mode.

```
Ruijie# configure terminal
Ruijie(config)#
```

- Enter ACL configuration mode.

```
Ruijie(config)# ip access-list extended test-tcp-flag
Ruijie(config-ext-nacl)#
```

- Add an ACE.

```
Ruijie(config-ext-nacl)# permit tcp any any match-all rst
Ruijie(config-ext-nacl)# permit tcp host 1.1.1.1 any established
```

- Add a deny ACE.

```
Ruijie(config-ext-nacl)# deny tcp any any match-all fin
```

- end

```
Ruijie(config-ext-nacl)# end
```

- Show

```
Ruijie# show  access-list test-tcp-flag
ip access-lists extended test-tcp-flag
10 permit tcp any any match-all rst
20 deny tcp any any match-all fin
```

Configuring Comments

Comments on ACLs and ACEs are provided for easy query and understanding of ACL configuration.

Note Up to one ACL comment and 2048 ACE comments can be configured in one ACL.

Caution The length of each comment is 100 bytes.

☑ The ACE comment is supported on the router only.

In privileged configuration mode, run the following commands to configure an ACL comment:

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **ip access-list standard** *id*	Enters ACL configuration mode.
Ruijie(config-std-nacl)# **list-remark** *comment*	Comments the ACL.

You can also run the following commands to set an ACL comment:

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **access-list** *id* **list-remark** *comment*	Sets the ACL comment.

In privileged configuration mode, run the following commands to configure an ACE comment:

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **ip access-list standard** *id*	Enters ACL configuration mode.
Ruijie(config-std-nacl)# **remark** *comment*	Comment the ACE.

You can also run the following commands to set an ACE comment:

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **access-list** *id* **list-remark** *comment*	Sets the ACE comment.

The following example shows how to configure the ACL comment and the ACE comment:

```
Ruijie(config)#ip access-list standard 1
Ruijie(config-std-nacl)#remark ace_remark_permit_62_start
Ruijie(config-std-nacl)#permit 192.168.197.62 0.0.0.0
Ruijie(config-std-nacl)#remark ace_remark_permit_62_end
Ruijie(config-std-nacl)#list-remark acl_remark_foo
Ruijie(config-std-nacl)#end
Ruijie#write
Ruijie#show access-lists 1
ip access-list standard 1
 remark ace_remark_permit_62_start
 10 permit host 192.168.197.62
 remark ace_remark_permit_62_end
list-remark acl_remark_foo
Ruijie#
```

Configuring SVI Router ACLs

The ACLs applied to layer 3 interfaces are called Router ACLs, which apply only to the routing packets forwarded at layer 3.

To solve this problem, Ruijie switches are configured with a command for enabling SVI Router ACLs. After this function is enabled, security ACLs on SVIs apply only to layer 3 forwarding packets between VLANs.

Default Configuration

By default, SVI Router ACLs are disabled. SVI ACLs apply to both inter-VLAN layer 3 packets and intra-VLAN bridge-forwarded packets.

Configuring SVI Router ACLs

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie# **[*no*] svi router-acls enable**	Enables/Disables the SVI Router ACLs.

☑ This function is only supported on SS3000E, S5750, S8600 and S12000 series routers.

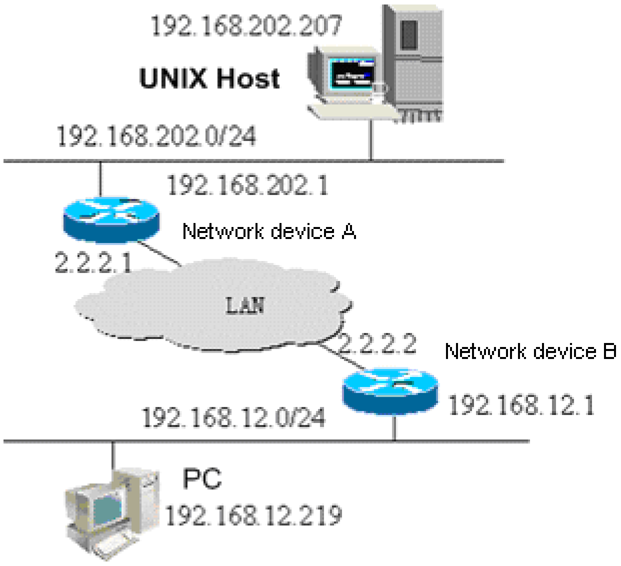
Configuration Examples

IP ACL Example

Configuration requirements:

There are two network devices A and B, as shown in Figure 1-3:

Figure 3 Basic ACL



It is required to implement the following security functions by configuring ACLs on device B.

Hosts on the 192.168.12.0/24 network segment can telnet the remote Unix host only in working hours and these host cannot ping the Unix server.

Device B is forbidden to access any services of hosts on the 192.168.202.0/24 network segment.

Note

This example shows a simplified topology of the banking system. Namely, only access from hosts on the LAN in a branch or outlet to the central host is allowed.

Equipment Configuration

Device B configuration:

```
Ruijie(config)# interface GigabitEthernet 0/1
Ruijie(config-if)# ip address 192.168.12.1 255.255.255.0
Ruijie(config-if)# exit
Ruijie(config)# interface GigabitEthernet 0/2
Ruijie(config-if)# ip address 2.2.2.2 255.255.255.0
Ruijie(config-if)# ip access-group 101 in
Ruijie(config-if)# ip access-group 101 out
```

According to requirements, configure an extended ACL numbered 101

```
access-list 101 permit tcp 192.168.12.0 0.0.0.255 any eq telnet time-range check
Ruijie(config)# access-list 101 deny icmp 192.168.12.0 0.0.0.255 any
Ruijie(config)# access-list 101 deny ip 2.2.2.0 0.0.0.255 any
Ruijie(config)# access-list 101 deny ip any any
```

Configure the time range

```
Ruijie(config)# time-range check
Ruijie(config-time-range)# periodic weekdays 8:30 to 17:30
```

Note

For ACL 101, the last rule statement ”access-list 101 deny ip any any” is not needed, because the end of the ACL contains an implicit deny statement for all packets.

Device A configuration:

```
Ruijie(config)# hostname Ruijie
Ruijie(config)# interface GigabitEthernet 0/1
Ruijie(config-if)# ip address 192.168.202.1 255.255.255.0
Ruijie(config)# interface GigabitEthernet 0/2
Ruijie(config-if)# ip address 2.2.2.1 255.255.255.0
```

IPv6 Extended ACL Configuration Example

It is required to implement the following security functions by configuring ACLs:

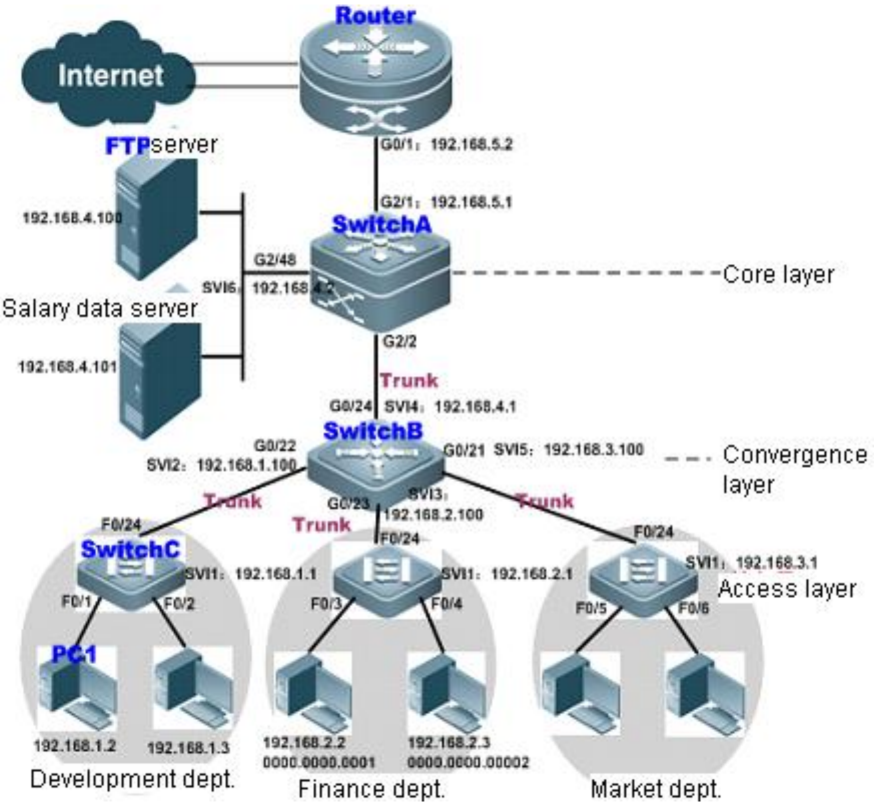
- The host whose IP address is 192.168.4.12 can access the gigabit 0/1 interface of a device.
- It cannot access other interfaces.

```
Ruijie> enable
Ruijie# config terminal
Ruijie(config)# ipv6 access-list v6-list
Ruijie(config-ipv6-nacl)# permit ipv6 ::192:68:4:12/24 any
Ruijie(config-ipv6-nacl)# deny ipv6 any any
Ruijie(config-ipv6-nacl)# exit
Ruijie(config)# interface gigabitEthernet 0/1
Ruijie(config-if)# ipv6 traffic-filter v6-list in
Ruijie(config-if)# end
Ruijie# show access-lists
ipv6 access-list extended v6-list
petmit ipv6 ::192.168.4.12 any
deny any any
```

☑ An ACL cannot match all the preceding areas. Besides, the IPv6 ACL does not apply to packet fragments. Besides, when **sip** and **dip** of a packet match an ACL, **type code** or source and destination ports of ICMP is ignored.

Typical Application of Intranet ACL

Figure 5 Networking Diagram



The preceding diagram shows the typical topology of an Intranet:

The access switch (Switch C) connecting PCs of respective departments is connected to the convergence switch through Gigabit optical cable (trunk mode).

The convergence switch (Switch B) assigns one VLAN for each department and is connected to the core switch through 10G optical fiber cable (trunk mode).

The core switch (Switch A) is connected with multiple servers, such as FTP, HTTP server and etc, and is connected to Internet through firewall.

Application Requirements

The ACL application in this network has the following requirements:

Ports that are susceptible to viruses must be disabled to guarantee Intranet security.

Only the internal PCs can access the servers.

Only PCs within a department can access each other.

R&D personnel are forbidden to use Instant messaging software such as QQ and MSN in working hours (namely from 09:00 to 18:00).

Notes

- The viruses can be avoided by configuring extended ACLs on the router-connecting port (G2/1) of core switch (Switch A) to filter packets destined for relevant ports.
- As for the requirement that internal PCs can access the servers while external PCs are not allowed to access these servers, IP extended ACLs can be defined and applied to ports (G2/2, SVI2) of the core switch (Switch A) that connect with the convergence switch and server.
- As for the requirement that specific departments cannot access each other, IP extended ACLs can be applied to G0/22 and G0/23 of Switch B).
- Configuring time & IP based extended ACL can prevent R&D departments from suing QQ/MSN and other IM application during a specific period (applying time & IP based extended ACL to SVI2 of Switch B).

Configuration Steps

- Configure the core switch: Switch A

Step 1: Define the virus-blocking ACL "Virus_Defence".

Note

The worm viruses on the network will create a TFTP server on the local port of "udp/69" in order to transmit the binary virus program to other infected systems. While selecting the destination IP address, the worms will generally select the IP address of subnet to which the infected system belongs, and then randomly select the attack target on Internet as per certain algorithm. Once the connection is established, the worms will send attack data to TCP ports (135, 445, 593, 1025, 5554, 9995, and 9996), UDP ports (136, 445, 593, 1433, and 1434) and UDP/TCP ports (135, 137, 138, and 139) of targets. If the attack is successful, TCP/4444 port of target system will be used as the backdoor port. After that, worms will connect to this port and send **tftp** command in order to transmit virus file to the target system and run the file. The infected server will send substantive invalid data packets to the network, thus wasting network bandwidth and even causing failure of network devices and the network. In such a case, the extended ACL can be used to filter data packets destined for these ports.

```
A#configure terminal
A(config)#ip access-list extended Virus_Defence
```

! Deny packets destined for internal and external TCP ports which may have been used by viruses.

```
A(config-ext-nacl)#deny tcp any any eq 135
A(config-ext-nacl)#deny tcp any eq 135 any
A(config-ext-nacl)#deny tcp any any eq 136
A(config-ext-nacl)#deny tcp any eq 136 any
A(config-ext-nacl)#deny tcp any any eq 137
A(config-ext-nacl)#deny tcp any eq 137 any
```

…………! The configuration on other ports is similar.

```
A(config-ext-nacl)#deny tcp any any eq 9996
A(config-ext-nacl)#deny tcp any eq 9996 any
```

! Deny packets destined for internal and external UDP ports which may have been used by viruses.

```
A(config-ext-nacl)#deny udp any any eq 69
A(config-ext-nacl)#deny udp any eq 69 any
A(config-ext-nacl)#deny udp any any eq 135
A(config-ext-nacl)#deny udp any eq 135 any
A(config-ext-nacl)#deny udp any any eq 137
A(config-ext-nacl)#deny udp any eq 137 any
```

! The configuration on other ports is similar.

```
A(config-ext-nacl)#deny udp any any eq 1434
A(config-ext-nacl)#deny udp any eq 1434 any
```

! Deny ICMP packets.

```
A(config-ext-nacl)#deny icmp any any
```

! Permit all other IP packets.

```
A(config-ext-nacl)#permit ip any any
A(config-ext-nacl)#exit
```

Step 2: Apply the ACL *Virus_Defence* to the router-connecting interface of the core device.

```
A(config)#interface gigabitEthernet 2/1
A(config-if)#no switchport
A(config-if)#ip address 192.168.5.1 255.255.255.0
```

! Apply the ACL *Virus_Defence* in the inbound direction of G2/1 to deny virus infected packets from an external network.

```
A(config-if)#ip access-group Virus_Defence in
A(config-if)#exit
```

Step 3: Define the ACL *access_server* that permits only Intranet PCs to access the server.

```
A(config)#ip access-list extended access_server
```

! Permit only specified Intranet PCs to access the server (IP address: 192.168.4.100).

```
A(config-ext-nacl)#permit ip 192.168.2.0 0.0.0.255 host 192.168.4.100
A(config-ext-nacl)#permit ip 192.168.1.0 0.0.0.255 host 192.168.4.100
A(config-ext-nacl)#permit ip 192.168.3.0 0.0.0.255 host 192.168.4.100
A(config-ext-nacl)#deny ip any any
```

Step 4: Apply the ACL *access_server* to the interface connecting with convergence device and server.

```
A(config)#interface gigabitEthernet 2/2
A(config-if)#switch mode trunk
```

! Apply the ACL to the inbound direction of the convergence switch.

```
A(config-if)#ip access-group access_server in
A(config-if)#exit
```

! Create a VLAN.

```
A(config)#vlan 2
A(config-vlan)#exit
A(config)#interface gigabitEthernet 2/48
```

! The server-connecting interface of G2/48 belongs to VLAN 2.

```
A(config-if)#switch access vlan 2
A(config-if)#exit
```

! Apply the ACL to the inbound direction of the server-connecting interface.

```
A(config)#interface vlan 2
A(config-if-VLAN 2)# ip access-group access_server in
A(config-if-VLAN 2)# ip address 192.168.4.2 255.255.255.0
A(config-ext-nacl)#end
```

- Configure the convergence switch: SwitchB

Step 1: Create VLAN 2, VLAN 3, and VLAN 4.

```
B#configure terminal
```

! Create VLAN 2, VLAN 3, and VLAN 4.

```
B(config)#vlan range 2-4
B(config-vlan-range)#exit
```

Step 2: Define ACLs.

! Define the IP extended ACLs vlan_access1 and vlan_access2.

```
B(config)#ip access-list extended vlan_access1
```

! Prohibit PCs of the finance department and market department from accessing PCs of the R&D department.

```
B(config-ext-nacl)#deny ip 192.168.2.0 0.0.0.255 192.168.1.0 0.0.0.255
B(config-ext-nacl)#deny ip 192.168.3.0 0.0.0.255 192.168.1.0 0.0.0.255
B(config-ext-nacl)#permit ip any any
B(config)#ip access-list extended vlan_access2
```

! Prohibit PCs of the R&D department and market department from accessing PCs of the finance department.

```
B(config-ext-nacl)#deny ip 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255
B(config-ext-nacl)#deny ip 192.168.3.0 0.0.0.255 192.168.2.0 0.0.0.255
B(config-ext-nacl)#permit ip any any
B(config-ext-nacl)#exit
```

Step 3: Apply ACLs *vlan_access1* and *vlan-access2* to the corresponding interfaces.

! Configure G0/22 as a trunk port and apply *vlan_access1* to this port.

```
B(config)#interface GigabitEthernet 0/22
B(config-if)#switchport mode trunk
B(config-if)#ip access-group vlan_access1 in
```

! Configure G0/23 as a trunk port and apply *vlan_access2* to this port.

```
B(config)# interface GigabitEthernet 0/23
B(config-if)# switchport mode trunk
B(config-if)# ip access-group vlan_access2 in
```

! Configure G0/24 as a trunk port.

```
B(config)#interface GigabitEthernet 0/24
B(config-if)#switchport mode trunk
```

! Configure the IP address of SVI 2.

```
B(config)#interface vlan 2
B(config-if)#ip address 192.168.1.100 255.255.255.0
```

! Configure the IP address of SVI 3.

```
B(config)#interface vlan 3
B(config-if)#ip address 192.168.2.100 255.255.255.0
```

! Configure the IP address of SVI 4.

```
B(config)#interface vlan 4
B(config-if)#ip address 192.168.4.1 255.255.255.0
```

Step 4: Define time range.

! Define the time range that starts from 09:00 to 18:00 in weekdays.

```
B#configure terminal
B(config)#time-range worktime
B(config-time-range)#periodic weekdays 9:00 to 18:00
```

Step 5: Define the traffic rule of R&D department.

```
B#configure terminal
```

! Create the extended ACL *yanfa* in configuration mode.

```
B(config)#ip access-list extended yanfa
```

! Prohibit all IM applications such as QQ and MSN on hosts of R&D department from 09:00 to 18:00 in weekdays.

```
B(config-ext-nacl)#deny tcp 192.168.1.0 0.0.0.255 eq 8000 any time-range worktime
B(config-ext-nacl)#deny tcp 192.168.1.0 0.0.0.255 eq 8001 any time-range worktime
B(config-ext-nacl)#deny tcp 192.168.1.0 0.0.0.255 eq 443 any time-range worktime
B(config-ext-nacl)#deny tcp 192.168.1.0 0.0.0.255 eq 1863 any time-range worktime
B(config-ext-nacl)#deny tcp 192.168.1.0 0.0.0.255 eq 4000 any time-range worktime
B(config-ext-nacl)#deny udp 192.168.1.0 0.0.0.255 eq 8000 any time-range worktime
B(config-ext-nacl)#deny udp 192.168.1.0 0.0.0.255 eq 1429 any time-range worktime
B(config-ext-nacl)#deny udp 192.168.1.0 0.0.0.255 eq 6000 any time-range worktime
B(config-ext-nacl)#deny udp 192.168.1.0 0.0.0.255 eq 6001 any time-range worktime
B(config-ext-nacl)#deny udp 192.168.1.0 0.0.0.255 eq 6002 any time-range worktime
B(config-ext-nacl)#deny udp 192.168.1.0 0.0.0.255 eq 6003 any time-range worktime
B(config-ext-nacl)#deny udp 192.168.1.0 0.0.0.255 eq 6004 any time-range worktime
```

! Permit all other IP traffic.

```
B(config-ext-nacl)#permit ip any any
```

! Apply the ACL to the inbound direction of SVI 2.

```
B(config)#interface vlan 2
B(config-if)#ip access-group yanfa in
```

Verification

Step 1: Verify whether ACEs are correct. The key is that whether the priorities of ACEs are correct and whether ACEs are effective.

```
SwitchA#show access-lists
ip access-list extended Virus_Defence
 10 deny tcp any any eq 135
 20 deny tcp any eq 135 any
 30 deny tcp any eq 4444 any
 40 deny tcp any any eq 5554
 50 deny tcp any eq 5554 any
 60 deny tcp any any eq 9995
 70 deny tcp any eq 9995 any
 80 deny tcp any any eq 9996
 90 deny tcp any eq 9996 any
 100 deny udp any any eq tftp
 110 deny udp any eq tftp any
 120 deny udp any any eq 135
 130 deny udp any eq 135 any
 140 deny udp any any eq netbios-ns
 150 deny udp any eq netbios-ns any
 160 deny udp any any eq netbios-dgm
 170 deny udp any eq netbios-dgm any
 180 deny udp any any eq netbios-ss
 190 deny udp any eq netbios-ss any
 200 deny udp any any eq 445
 210 deny udp any eq 445 any
 220 deny udp any any eq 593
 230 deny udp any eq 593 any
 240 deny udp any any eq 1433
 250 deny udp any eq 1433 any
 260 deny udp any any eq 1434
 270 deny udp any eq 1434 any
 280 deny tcp any any eq 136
 290 deny tcp any eq 136 any
 300 deny tcp any any eq 137
 310 deny tcp any eq 137 any
 320 deny tcp any any eq 138
 330 deny tcp any eq 138 any
 340 deny tcp any any eq 139
 350 deny tcp any eq 139 any
 360 deny tcp any any eq 445
 370 deny tcp any eq 445 any
 380 deny tcp any any eq 593
 390 deny tcp any eq 593 any
 400 deny tcp any eq 1025 any
 410 deny tcp any any eq 4444
420 deny icmp any any
 430 permit tcp any any
 440 permit udp any any
 450 permit ip any any

ip access-list extended access_server
 10 permit ip 192.168.2.0 0.0.0.255 host 192.168.4.100
 20 permit ip 192.168.1.0 0.0.0.255 host 192.168.4.100
 30 permit ip 192.168.3.0 0.0.0.255 host 192.168.4.100
 40 deny ip any any
SwitchB#show access-lists
ip access-list extended vlan_access1
 10 deny ip 192.168.2.0 0.0.0.255 192.168.1.0 0.0.0.255
 20 deny ip 192.168.3.0 0.0.0.255 192.168.1.0 0.0.0.255
 30 permit ip any any

ip access-list extended vlan_access2
 10 deny ip 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255
 20 deny ip 192.168.3.0 0.0.0.255 192.168.2.0 0.0.0.255
 30 permit ip any any

ip access-list extended yanfa
 10 deny tcp 192.168.1.0 0.0.0.255 eq 8000 any time-range worktime (active)
 20 deny tcp 192.168.1.0 0.0.0.255 eq 8001 any time-range worktime (active)
 30 deny tcp 192.168.1.0 0.0.0.255 eq 443 any time-range worktime (active)
 40 deny tcp 192.168.1.0 0.0.0.255 eq 1863 any time-range worktime (active)
 50 deny tcp 192.168.1.0 0.0.0.255 eq 4000 any time-range worktime (active)
 60 deny udp 192.168.1.0 0.0.0.255 eq 8000 any time-range worktime (active)
 70 deny udp 192.168.1.0 0.0.0.255 eq 1429 any time-range worktime (active)
 80 deny udp 192.168.1.0 0.0.0.255 eq 6000 any time-range worktime (active)
 90 deny udp 192.168.1.0 0.0.0.255 eq 6001 any time-range worktime (active)
 100 deny udp 192.168.1.0 0.0.0.255 eq 6002 any time-range worktime (active)
 110 deny udp 192.168.1.0 0.0.0.255 eq 6003 any time-range worktime (active)
 120 deny udp 192.168.1.0 0.0.0.255 eq 6004 any time-range worktime (active)
```

Step 2: Verify whether ACL configuration is complete. The key is that whether the correct ACL has been applied to the specified interface.

Device A configuration:

```
A#show run
interface GigabitEthernet 2/1
 no switchport
 no ip proxy-arp
 ip access-group Virus_Defence in
 ip address 192.168.5.1 255.255.255.0
!
interface GigabitEthernet 2/2
 switchport mode trunk
 ip access-group access_server in
!
interface VLAN 2
 no ip proxy-arp
 ip access-group access_server in
 ip address 192.168.4.2 255.255.255.0
```

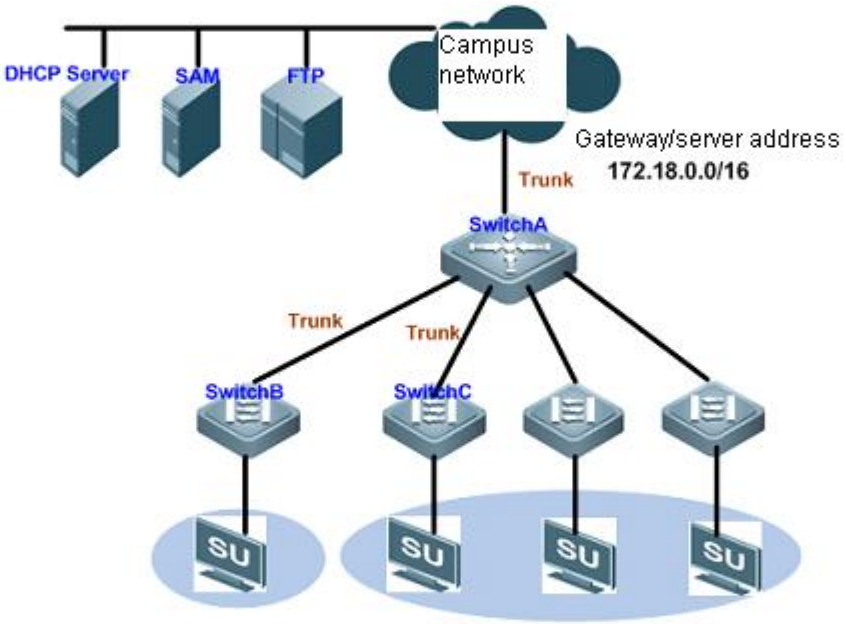
Device B configuration:

```
B#show run
!
interface GigabitEthernet 0/22
 switchport mode trunk
 ip access-group vlan_access1 in
!
interface GigabitEthernet 0/23
 switchport mode trunk
 ip access-group vlan_access2 in
!
interface VLAN 2
 no ip proxy-arp
 ip access-group yanfa in
 ip address 192.168.1.100 255.255.255.0
```

Application of expert ACL & ACL 80

Networking Diagram

Figure 6 Application topology diagram of the expert ACL&ACL 80



The preceding figure shows the simplified topology of a campus network:

Switch A is the convergence device assigning one VLAN for each faculty and is connected to the campus network through 10G optical cable (trunk mode).

Switch B and Switch C are access devices connecting PCs of respective faculties, and are connected to the convergence switch through Gigabit optical cable (trunk mode).

SU client must be installed on each PC so that a PC can access Internet after being authenticated.

Application Requirements

SU software is not embedded in Windows. You must download and install SU client on the PC. However, the PC cannot download software before 802.1x authentication. To solve this problem, the following requirements must be met:

- IP packets and ARP packets accessing the segment address of gateway/server (172.18.0.0/16) are allowed to pass through without authentication, so that the user PC can download software from the specified server or access gateway before authentication.
- DHCP packets (UDP port number being 67/68) are allowed to pass through without authentication, so that the user PC can acquire the IP address in order to proceed with authentication.

Notes

Configure ACL80 or expert ACL on the access device (Switch B/Switch C) and combine the feature of secure tunnel to permit certain packets without authentication.

In this case, ACL 80 is configured on Switch B and expert ACLs are configured on Switch C.

Configuration Steps

- Device B configuration

Note

ACL 80 allows the user to define 16 bytes out of the first 80 bytes of packets to perform per-bit matching and filtering. The user-defined string will be compared with the string extracted from packet (1 means match and 0 means mismatch), so as to determine further action.

Step 1: Configure the customized ACL.

```
B#configure terminal
```

! Create a customized ACL named "tongdao"

```
B(config)#expert access-list advanced tongdao
```

! Permit all ARP packets (protocol number being 0806, offset being 24) with source IP address(the offset in the source IP of ARP packets is 40) falling within the network segment of 172.18.0.0 (hexadecimal value being ac12)

```
B(config-exp-dacl)#permit 0806 ffff 24 ac12 ffff 40
```

! Permit all IP packets (protocol number being 0800, offset being 24) with source IP (the offset in the source IP of IP packets is 38) falling within the network segment of 172.18.0.0 (hexadecimal value being ac12)

```
B(config-exp-dacl)#permit 0800 ffff 24 ac12 ffff 38
```

! Permit DHCP packets with UDP port being 67 (Bootstrap Protocol Server) and 68 (Bootstrap Protocol Client) (offset in protocol number being 35; hexadecimal value of 11 to indicate UDP; offset in port being 46; hexadecimal value of 43/44 corresponding to 67 and 68).

```
B(config-exp-dacl)# permit 11 ff 35 00440043 ffffffff 46
B(config-exp-dacl)#exit
```

Step 2: Globally configure the ACL for secure tunnel application.

! Configure ACL "tongdao" for secure tunnel application

```
B(config)# security global access-group tongdao
```

- Device C configuration:

Step 1: Configure an expert ACL.

```
C#configure terminal
```

! In configuration mode, create an expert ACL named "tongdao1"

```
C(config)#expert access-list extended tongdao1
```

! Permit all IP packets with source IP falling within the network segment of 172.18.0.0

```
C(config-exp-dacl)#permit ip 172.18.0.0 0.0.255.255 any any any
```

! Permit all packets with UDP port number being 67 (Bootstrap Protocol Server) and 68 (Bootstrap Protocol Client)

```
C(config-exp-dacl)# permit udp any any eq bootpc any any eq bootps
```

```
C(config-exp-dacl)#exit
```

Step 2: Globally configure the ACL for secure tunnel application.

! Configure ACL "tongdao1" for secure tunnel application

```
C(config)# security global access-group tongdao1
```

Verifications

Step 1: Verify whether ACEs are correct. The key is that whether the priorities of ACEs are correct and whether ACEs are effective.

```
B# show access-lists
expert access-list advanced tongdao
 10 permit 0806 FFFF 24 AC12 FFFF 40
 20 permit 0800 FFFF 24 AC12 FFFF 38
 30 permit 11 FF 35 00440043 FFFFFFFF 46
C# show access-lists
expert access-list extended tongdao1
 10 permit ip 172.18.0.0 0.0.255.255 any any any
 20 permit udp any any eq bootpc any any eq bootps
```

Run the preceding command to verify whether the corresponding ACEs are correct.

Step 2: Verify whether ACL configuration is complete. The key is that whether the correct ACL has been applied in global configuration mode:

```
B#show run
!
expert access-list advanced tongdao
!
security global access-group tongdao
!
!
C#show run
!
expert access-list advanced tongdao1
!
security global access-group tongdao1
!
```

☑ ACL configuration for different line cards:

The following description applies only to versions later than RGOS10.3.

This principle is also appropriate for hot pluggable line cards, which prompts the users to reset line cards.

If ACL out is implemented on the egress, then IP extended ACL and expert ACL will not support port matching. Besides, expert ACL only supports IP packet matching, not other L2 packets, IPV6 does not support flow_label, DSCP and fragment matching.

If ACL out is processed in the original way, then associating ACL out with SVIs has lots of restrictions:

- Changes the priority of in and out direction; the ACL used in the outbound direction is higher than that used in the inbound direction.
- When you apply an ACL to the outbound direction of an SVI, there is no **deny any any** option by default. But there is **deny any any** option in other ACLs.
- Associating ACL with SVI in Out direction can support IP standard, IP extended, MAC extended, ACL application of expert extended ACLs.
- There are some restrictions for matching destination ip and destination mac in ACL when associating ACL with SVI in the outbound direction. If you want to match destination MAC in MAC extended and expert ACL and apply the ACL in the outbound direction of SVI, the entry will be set and not take effect.
- The set ACL will not take effect if you want to match destination IP address, which is not within the subnet IP range of associated SVI, in IP standard, IP extended and expert ACL. For example, the address of VLAN 1 is 192.168.64.1 255.255.255.0. And now, if you create an IP extended ACL with ACE deny udp any 192.168.65.1 0.0.0.255 eq 255, it will not take effect when applying this ACL to the egress of VLAN 1, for the destination IP address is not within the subnet IP range of VLAN 1; but it will take effect if the ACE is deny udp any 192.168.64.1 0.0.0.255 eq 255, for the destination IP address is up to specification.
- The priority of the ACL associated with an SVI in the outbound direction has the highest priority.
- ACL out does not support user-defined acl type.

Configuring the Firewall

Understanding IP-MAC Binding

Overview

IP-MAC binding refers to that the IP address and MAC address of a host are bound on the router or firewall it directly connects to, so that an IP address can be used only by the host with the matching MAC address. This function is designed to prevent IP address spoofing. To deploy this function, two prerequisites must be met:

1) The MAC address is unique and genuine.
2) IP-MAC binding applies only to hosts that are directly connected to a router or firewall.

Furthermore, a host interface may be configured with multiple IP addresses, thus allowing multiple IP addresses to be bound to the same MAC address.

Configuring IP-MAC Binding

Enabling or Disabling the IP-MAC Binding Function

IP-MAC binding is disabled by default. To use this function, configure binding rules by using the **ipmacbind** command in global configuration mode. IP-MAC binding function will be disabled if all IP/MAC binding rules are deleted.

Configuring IP-MAC Binding

The IP address of a host can be bound to its MAC address by using the **ipmacbind** command to prevent the IP address from being counterfeited by other hosts. Rebind the bound IP address if it already exists. Multiple IP addresses can be bound to the same MAC address. For example:

```
Ruijie(config)# ipmacbind 192.168.52.69 032a.33ac.3f11 log
```

The preceding command binds the IP address 192.168.52.69 to the network card with MAC address 032a.33ac.3.3f11. Here, **log** indicates that the log function regarding IP-MAC binding is enabled. Besides, you can specify the format of any IP address and any MACaddress. For example:

Ruijie(config)# **ipmacbind any any log**

Meanwhile, the IP-MAC binding entries on a LAN can be detected and exported from the ARP table dynamically by using the **ipmacbind auto** command.

```
Ruijie(config)# ipmacbind auto
```

You can configure an IP-MAC binding rule list as well as the rules in the list and apply the list to the interface. Besides, you can specify the default processing of packets not matching the IP-MAC binding rule on the current interface. For example:

```
Ruijie(config)# ipmacbind list number
Ruijie(config-ipmac-bind)#ipmacbind ip mac [log]
Ruijie(config-if-GigabitEthernet 0/0)# ipmacbind list number [ default action {permit | deny
[ log ]} ]
```

Configure the IP MAC binding rule list and the corresponding rules in the list, apply the list to the corresponding interface at the same time, and configure the default rule operation on the current interface.

```
Ruijie(config)# ipmacbind list number
Ruijie(config-ipmac-bind)#ipmacbind ip mac [log]
Ruijie(config-if-GigabitEthernet 0/0)# ipmacbind list number [ default action {permit | deny
[ log ]} ]
```

The **no** form of the **Ipmacbind** command is used to delete IP-MAC binding entries. If an IP address is specified in this command, the IP address is deleted from an IP-MAC binding entry. Besides, you can use the **clear ipmacbind** command to delete the IP-MAC binding entries dynamically exported from the ARP table or all IP-MAC binding entries. For example:

```
Ruijie# clear ipmacbind dynamic
```

The preceding command clears all IP-MAC binding entries dynamically exported from the ARP table by using the **ipmacbind auto** command.

```
Ruijie# clear ipmacbind all
```

The preceding command clears all IP-MAC binding entries including the information of the rule list.

By default, packets without IP MAC address binding rule are permitted to pass. Use the ipmacbind default action command to deny packets without IP MAC address.

```
Ruijie(config)#ipmacbind default action deny
```

Rule description:

The IPMAC binding rule operation only permits or denies two situations. By default, the rule is permit, and the matching rule combinations and the results are as follows:

Deny:

IP Address	MAC Address	Result
False	Correct	Deny
Correct	False	Deny
False	False	Deny
Correct	Correct	Permit

Permit:

IP Address	MAC Address	Result
Correct	False	Deny
Correct	Correct	Permit
False	Correct	Permit
False	False	Permit

Viewing IP-MAC Binding Information

Run the **show ipmacbind** command to view records or statistics on current IP-MAC binding.

```
Ruijie# show ipmacbind table
Totol number of IPMAC-Bind rule: 2
IPMAC-Bind global rule:
No    Type        IP Address     MAC Address     Log
1     <static>    any            00d0.0011.0012   off

IPMAC-Bind list 1 rule:
No     Type        IP Address      MAC Address     Log
1    <static>    192.168.2.2     00d0.0011.0011   off
```

To view all IP-MAC binding entries and the IP-MAC binding rule information:

```
Ruijie# show ipmacbind statistic
IPMAC-Bind global dropped 0 packets
IPMAC-Bind list 1 dropped 0 packets
```

To view the IP MAC function and the number of invalid packets intercepted by the IP MAC binding rule list;

```
Ruijie# show ipmacbind hash
IPMAC-Bind global:
In MAC hash-list 211:
   1: ip-any, mac-00d0.0011.0012
IPMAC-Bind list 1:
In IP hash-list 616:
1: ip-192.168.2.2, mac-00d0.0011.0011
```

To view the IP-MAC binding rule and the hash table corresponding to IP-MAC binding rule list:

Understanding URL Filtering

Overview

URL filtering is an extension of packet filtering, achieving in-depth access control. It is a means of content filtering and restricts intranet users from accessing certain illegal websites.

The process of URL filtering is as follows:

1) The firewall resolves the HTTP request from a client to obtain the requested URL;
2) The firewall uses the predefined URL filtering rules to match the requested URL;
3) If a match is detected, the firewall determines whether to permit or reject the access based on the matching result.
4) If no match is detected, the firewall sends the URL request to a third-party content filtering server (such as Websense or N2H2) and meanwhile suspends the HTTP session until a response is returned from the server. Then the firewall determines whether to permit or reject the HTTP request based on the response. All requests shall be denied and an alarm is reported if the server is unavailable.

Currently, only local URL filtering is supported. The linkage with a third-party content filtering server will be realized in future.

Configuring URL Filtering

URL filtering rules are configured as follows:

5) Specify the URLs to be filtered and add these URLs to a filtering category.
6) Configure filtering rules and add the filtering category to the rules.
7) Apply these rules to interfaces.

It should be noticed that you need to specify an ACL before applying the rules to the interfaces for advertising the URL filtering range.

Registering URLs

To filter a URL, use a URL-related command in global configuration mode to register the URL and the filtering category the URL belongs to. To delete a registered URL, run the **no url** command.

It should be noticed that a URL always begins with a stop (.) in all cases.

For example:

```
Ruijie(config)# ip urlfilter rule test .sina.com.cn
Ruijie(config)# ip urlfilter rule test .*sina.com.cn
Ruijie(config)# ip urlfilter rule test .sina*
Ruijie(config)# ip urlfilter rule test .*sina*
```

In the example, the asterisk (*) is used as the wildcard. The wildcard can be placed only at the end of a URL or after the first stop in a URL.

These URL formats are explained as follows:

- .sina.com.cn represents URLs such as www.sina.com.cn, blog.sina.com.cn, and new.sina.com.cn, instead of Ruijie.blog.sina.com.cn, sports.news.sina.com.cn, and the like.
- .*sina.com.cn represents all URLs ended with "sina.com.cn".
- .sina* represents all URLs in which the first stop is followed by "sina". For example, blog.sina.com.cn matches this rule but Ruijie.blog.sina.com.cn does not.
- *sina* represents all URLs containing "sina" such as Ruijie.blog.sina.com.cn, blog.sina.com.cn, and www.adfsina.com.

Configuring URL Filtering Rules

Filtering rules should be configured after the URL category is configured.

URL filtering rules are configured in global configuration mode.

```
Ruijie(config)# ip urlfilter category 1 test
```

The preceding command is to create a filtering rule numbered 1. Then, the predefined filtering category can be added to this rule. Each rule can accommodate 15 categories. That is, other predefined categories can also be added to this rule. For example:

- Ruijie(config)# ip urlfilter category 1 test1
- Ruijie(config)# ip urlfilter category 1 test2
- Ruijie(config)# ip urlfilter category 1 test3

It should be noticed that Ruijie products allow a category to be added to different rules.

```
Ruijie(config)# ip urlfilter category 1 test
Ruijie(config)# ip urlfilter category 2 test
Ruijie(config)# ip urlfilter category 3 test
```

Applying Rules to Interfaces

After the preceding configuration is complete, you need to apply these rules to interfaces.

Before applying rules to interfaces, remember configuring an ACL rule in configuration mode.

The following example shows how to create a URL that filters all IP addresses and allows access to the URLs defined in rule 1. In this example, logs are recorded when access is denied.

```
Ruijie(config)# access-list 1 permit any
Ruijie(config)# interface gigabitEthernet 0/0
Ruijie(config-if)# ip urlfilter exclusive-domain 1 1 permit in log
```

Now, URL filtering rule configuration is complete.

Viewing Configuration Information and Statistical Information of the URL Filtering Module

Configuration Information

Suppose that some URL filtering rules have been configured. The following describes how to query, configure, and delete these rules,

show ip urlfilter config address

Run the preceding command in configuration mode.

```
Ruijie(config)# show ip urlfilter config address
================[Url without wildcard]===================
cls_name          cls-id           url-address
=========================================================
test1              2                .tianya.cn
----------------------------------------------------------
test2             3                .sohu.com
----------------------------------------------------------
test1             1                .sina.com.cn
----------------------------------------------------------
test1              2                .mop.com
=================[Url no-wildcard end]====================
===================[Url with pre-wildcard]================
cls_name          cls-id           url-address
```

```
==========================================================
test1              2                .*hong.com
-----------------------------------------------------------
test2             3                .*263.net
-----------------------------------------------------------
test1              2                .*163.com
===================[Url pre-wildcard end]===================
===================[Url with post-wildcard]=================
cls_name          cls-id            url-address
===========================================================
test2             3                .taob*
-----------------------------------------------------------
test1              2                .google.com*
-----------------------------------------------------------
test2             3                .ebay*
-----------------------------------------------------------
test1              2                .baid*
===================[Url post-wildcard end]==================
===================[Url with all-wildcard]==================
cls_name          cls-id            url-address
===========================================================
test2             3                .*huawei*
-----------------------------------------------------------
test1              2                .*cisco*
===================[Url all-wildcard end]===================
=================[Relative CLI Command]=====================
ip urlfilter rule test1 .tianya.cn
ip urlfilter rule test2 .sohu.com
ip urlfilter rule test .sina.com.cn
ip urlfilter rule test1 .mop.com
ip urlfilter rule test1 .*hong.com
ip urlfilter rule test2 .*263.net
ip urlfilter rule test1 .*163.com
ip urlfilter rule test2 .taob*
ip urlfilter rule test1 .google.com*
ip urlfilter rule test2 .ebay*
ip urlfilter rule test1 .baid*
ip urlfilter rule test2 .*huawei*
ip urlfilter rule test1 .*cisco*
==========[Relative CLI Command To Del the Rules ]==========
no ip urlfilter rule test1 .tianya.cn
no ip urlfilter rule test2 .sohu.com
no ip urlfilter rule test .sina.com.cn
no ip urlfilter rule test1 .mop.com
no ip urlfilter rule test1 .*hong.com
no ip urlfilter rule test2 .*263.net
no ip urlfilter rule test1 .*163.com
no ip urlfilter rule test2 .taob*
no ip urlfilter rule test1 .google.com*
no ip urlfilter rule test2 .ebay*
no ip urlfilter rule test1 .baid*
no ip urlfilter rule test2 .*huawei*
no ip urlfilter rule test1 .*cisco*
```

Firstly, classify the addresses into four categories: addresses without a wildcard, addresses preceded by a wildcard, addresses followed by a wildcard and addresses with one wildcard at the beginning and one wildcard at the end. And we print out the classified addresses under the same category in a continuous manner.

Secondly, record all the commands you use to configure these addresses. As a result, even a green hand will be able to view the configuration process.

Lastly, provide you a cookie that shows you the commands for deleting configuration, so that you no longer need to run **show run** to query the command for deleting a configuration but copy the desired command and paste it in the CLI.

show ip urlfilter config rule

The **show** command shows the addresses that have been configured in the current system. Now let's learn the configured rules.

```
Ruijie(config)# show ip urlfilter config rule
=================[ Ip UrlFilter Rule configure ]=================
Id     Attribute         Details
-------------------------------------------------------------------------==--------------
--------
1      contain-class:     test
ref-interface:gigabitEthernet 0/0 gigabitEthernet 0/1
---------------------------------------------------------------------------==------------
---------
2      contain-class:     test1
ref-interface:     gigabitEthernet 0/0
gigabitEthernet 0/1
gigabitEthernet 0/2
-----------------------------------------------------------------------------------------
----------
3      contain-class:     test2
ref-interface:     gigabitEthernet 0/2
==============================================================
================[Relative CLI Command]========================
ip urlfilter category 1 test
ip urlfilter category 2 test1
ip urlfilter category 3 test2
=========[Relative CLI Command To Del the Rules ]===============
no ip urlfilter category 1 test
no ip urlfilter category 2 test1
no ip urlfilter category 3 test2
Using this command, you can view the address categories included in a rule and the interface
to which rules apply. Then you can determine the interface from which the desired information
can be obtained.
```

show ip urlfilter config setting

Query information about gigabitEthernet 0/0.

```
Ruijie(config-if)# show ip urlfilter config setting
========[ Url Filter Rules On gigabitEthernet 0/0 ]==============
Rules On Input
==============================================================
Id  Acl   Action  Class-name     Url-address
-----------------------------------------------------------------------------------------
--------
1   1    permit  test         .sina.com.cn
-----------------------------------------------------------------------------------------
--------
2   12   permit  test1          .tianya.cn
                              .mop.com
                              .*hong.com
                              .*163.com
                              .google.com*
                              .baid*
                              .*cisco*
-----------------------------------------------------------------------------------------
--------
3   12   block   test2          .sohu.com
                              .*263.net
                              .taob*
                              .ebay*
                              .*huawei*
-----------------------------------------------------------------------------------------
-------
2   13   permit  test1          .tianya.cn
                              .mop.com
                              .*hong.com
                              .*163.com
                              .google.com*
                              .baid*
                              .*cisco*
-----------------------------------------------------------------------------------------
--------
3   13   block   test2          .sohu.com
                              .*263.net
                              .taob*
                              .ebay*
                              .*huawei*
==========================================================
Relative CLI Command
==========================================================
ip urlfilter exclusive-domain 1 1 permit in log
ip urlfilter exclusive-domain 2 12 permit in
ip urlfilter exclusive-domain 3 12 block in
ip urlfilter exclusive-domain 2 13 permit in
ip urlfilter exclusive-domain 3 13 block in
-----------------------------------------------------------------------------------------
--------
Relative CLI Command to Del Rules
-----------------------------------------------------------------------------------------
--------
no ip urlfilter exclusive-domain 1 1 permit in log
no ip urlfilter exclusive-domain 2 12 permit in
no ip urlfilter exclusive-domain 3 12 block in
no ip urlfilter exclusive-domain 2 13 permit in
no ip urlfilter exclusive-domain 3 13 block in
======[ Url Filter Rules On gigabitEthernet 0/0 End]=======
```

In the command output, you can view rule IDs, effective scope of rules (ACL numbers), conform-action, inclusive categories, and URLs in these categories.

Statistical Information

Statistical information is also what administrators care much for.

```
Ruijie(config)# show ip urlfilter statistics
url filter statistics
==============================================
the rule 1
       Total requests allowed: 0
       Total requests blocked: 0
the rule 2
       Total requests allowed: 0
       Total requests blocked: 0
the rule 3
       Total requests allowed: 0
       Total requests blocked: 0
```

Problems Encountered in Use of this Function

By now, you have seen that URL filtering function configuration is very flexible. But flexible things are always hard to command. Next, let's together have a look at some matters that require your attention.

One URL can only belong to one address category at one time.

Just as you have seen, one URL can only belong to one rule at one time according to our rules.

Only one rule applies to one interface.

Perhaps you have noticed that we provide only one action for each rule in regard to the rules for interfaces, i.e., the action we take when the website to access to matches our rules. What about the mismatch? Take opposite action, of course.

Different URLs have different priorities in the matching process.

Rules' priorities are arranged as follows in descending order:

- Rules without a wildcard.
- Rules with a wildcard at the beginning.
- Rules with a wildcard at the end
- Rules with one wildcard at the beginning and the other at the end.

Once matched by a rule with high priority, an address will not be matched by following rules.

```
ip urlfilter rule test .sina.com.cn
ip urlfilter rule test1 .*.*

ip urlfilter category 0 test
ip urlfilter category 1 test1
!
!
ip access-list standard 1
10 permit any

interface gigabitEthernet 0/0
ip urlfilter exclusive-domain 1 1  block in log
ip ref
ip address 130.130.130.1 255.255.255.0
duplex auto
speed auto
!
interface gigabitEthernet 0/1
ip ref
ip address 192.168.52.141 255.255.255.0
duplex auto
speed auto

ip route  0.0.0.0 0.0.0.0  192.168.52.1
```

What is the function of this rule?

He has only configured two rules: one is for filtering .sina.com.cn, and the other is for filtering all websites with the character "." What did he do next? He added the two categories to the two rules, and apply the rule that filters all websites with the character "." to the ingress. It seems that he really wants to disable Internet access.

Well, guess whether the current rule can filter all websites or not? The answer is: of course not.

Take it easy. I will answer your question.

In order to save time and space, I will directly tell you that those websites with .sina.com.cn will not be filtered. Why?.

Let's analyze the process of searching, matching and filtering.

First of all, the website that users visit will match in the rule without wildcard. If they access www.sina.com.cn or news.sina.com.cn, the website will match the rule. Notice that once matched by any rule, the URL a user visits will no longer match following rules. This shall be emphasized. Then, as for our filtering program, the visited website falls into the category of "test" instead of "test 1".

When the category is found out, the filtering begins. Our filtering program on the interface will search in its own rule. When it finds that the category "test" is not included in its own rule, it will adopt the processing mode opposite to what is defined in its own rule to deal with this website. Since the guy defined the rule action is 'block", the action taken for the category "test" that is not in his rule will surely be "permit".

Now, let's have another look from the very beginning before we come to the conclusion why this guy has not filtered all rules. It is because he has not noticed the priority problem of website matching. Though the rule .*.* can match all addresses, .sina.com.cn that has a higher priority will match in a better way.

Therefore, we can say that it is very important to configure rule priority.

Notice should be given to the website with lower level domain name(s) and URL redirection.

When we verified our own products, our filter device was always disabled for some websites. Later, we kept track of the visits to these websites. The result was really surprising. Let's see the example of www.google.com.

Careful users may have found that when they access Google in China, they use www.google.com, but they come to see www.google.cn in the address bar of browser when the webpage is opened. This is called redirection.

However, under a real environment, the access process is more complicated. (Take our environment here for example).

First, we resolve the domain name of Google in DNS. Then, we will access the website of 210.70.14.147. Next, we will visit www.google.cn.

If you want to deal with such websites, you need to add the IP address and google.cn into the rule.

Then, you can use tailored functions as you wish.

Understanding Network Ingress Filtering

Overview

A lot of DoS/DDoS attacks are carried out with forged source IP addresses. NIF (Network Ingress Filtering, RFC 2827) is aimed to defend against such attacks, or limit the scope and lower the risk of being attacked. It will check up whether the source IP address claimed by the data packet entering a network meets the network prefix advertised by route. If not, filter it. Such filtering mechanism implemented on the router at the network ingress will be very effective to prevent IP spoofing attacks. However, it will take no effect on the IP spoofing attacks with legitimate IP address prefix at all.

Configuring NIF

Enabling or Disabling Network Ingress Filtering

NIF is disabled by default. To use this function, run the **ip ingress -filter** command in interface mode. For example:

```
Ruijie(config-if)# ip ingress-filter log
```

The preceding command enables the NIF function on the interface and the log function of NIF at the same time.

The **no** form of **ip ingress-filter** disables the NIF function on the interface. For example:

```
Ruijie(config-if)# no ip ingress-filter log
```

The preceding command disables the NIF function on the interface.

Viewing Network Ingress Filtering Information

Run the **show ip ingress-filter** command to view current IP/MAC binding records or statistical information, e.g.,.

```
Ruijie(config)# show ip ingress-filter
Firewall Network-ingress-filter is enable, blocked 0 flowsInterface FastEthernet 1/0: log is
on, blocked 0 flows
```

Through the preceding command, you can view whether the NIF function is enabled and how much unauthorized information has been blocked.

Understanding TCP SYN Proxy

Overview

SYN proxy is an effective way to guard against SYN Flood attacks. SYN flood attacks occur at the stage of three-way handshakes of TCP. Attackers send a large number of TCP SYN packets to victims, who then open a lot of TCP links and respond attackers by sending TCP SYN ACK. However, attackers do not send TCP ACK packets to complete three-way handshakes. In this case, thus victims' queues are filled with semi-connections and new connections cannot be established until these semi-connections time out. The basic process of TCP SYN proxy is that three-way handshakes between router/firewall proxy service terminal and client terminal are completed first, if the connection is legal, then connection with the service terminal will be established.

Configuring TCP SYN Proxy

Enabling or Disabling TCP SYN Proxy Function

TCP SYN proxy function is disabled by default. Before enabling this function, configure ACL rules to specify the streams to which TCP SYN proxy applies, e.g.,.

```
Ruijie(config)# access-list 100 permit ip 192.168.52.0 0.0.0.255 any
```

Then, run the **ip tcp-intercept list in|out** command in interface configuration mode to apply TCP SYN proxy to the streams that pass through the interface and match the ACL rules, e.g.,.

```
Ruijie(config)# interface gigabitEthernet 0/0
Ruijie(config-if)# ip tcp-intercept list 100 in log
```

The preceding command indicates that TCP SYN proxy is applied to the inbound streams of gigabit0/0 complying with ACL 100 and the log function of TCP SYN proxy is enabled.

Use the **no** form of the **ip tcp-intercept list in|out** command to disable this function, e.g.,.

```
Ruijie(config)# interface gigabitEthernet 0/0
Ruijie(config-if)# no ip tcp-intercept list 100 in log
```

Viewing TCP SYN Proxy Information

Use the **show ip tcp-intercept** command to view current TCP SYN proxy records or statistical information, e.g.,.

```
Ruijie(config-if)# show ip tcp-intercept
Intercepting new connections using access-list 100 at gigabitEthernet 0/0 in
20 incomplete, 1320 established connections (total 1340)
```

Through the preceding command, you can see that TCP SYN proxy has been enabled on gigabit 0/0 interface. In addition, 20 invalid TCP streams unfinished and 1,320 normal TCP streams have been monitored.

Understanding Special Protocol

Overview

Such protocols as FTP, MMS, H.323, etc have separate command control channels and data channels. And the data channels are port numbers, etc randomly specified through the control channels by both channels. If control channel port access is only allowed by configuring ACL and other packet filtering rules on network equipments, and provided that no special means is available for processing, data channels will be completely blocked. Therefore, a special way is needed to establish some temporary pass and access mechanisms for the data channels of these protocols.

Configuring a Special Protocol

Users need to configure rules in configuration mode, where they specify a name that consists of character strings and is easy to remember for a special protocol to access, e.g.,.

You can add FTP to the rule library named "abc", to which the MMS protocol can also be added. These two protocols do not overlay or conflict with each other.

In the same way, a rule library named "123" may be defined.

```
Ruijie(config)# ip inspect name abc ftp
Ruijie(config)# ip inspect name abc mms
Ruijie(config)# ip inspect name 123 mms
Ruijie(config)# ip inspect name 123 h323

Ruijie(config)# show ip inspect all
Inspection Rule Configuration
Inspection name abc
```

```
ftp
mms
Inspection name 123
mms
h323
```

Then, users need to enter the specified interface and add this rule. For example:

```
Ruijie(config)# interface gigabitEthernet 0/0
Ruijie(config-if)# ip inspect abc in
Ruijie(config)# show ip inspect all
Inspection Rule Configuration
Inspection name abc
ftp
mms
Inspection name 123
mms
h323

Interface Configurationn
Interface gigabitEthernet 0/0
Inbound inspection rule is abc
ftp
mms
```

It should be noticed that one interface can only be applied with one special protocol rule library. If added with another, the newly added rule shall replace the original one. For example:

```
Ruijie(config)# interface gigabitEthernet 0/0
Ruijie(config-if)# ip inspect abc in
Ruijie(config)# show ip inspect all
Inspection Rule Configuration
Inspection name abc
ftp
mms
Inspection name 123
mms
h323
Interface Configurationn
Interface gigabitEthernet 0/0
Inbound inspection rule is abc
ftp
mms
Ruijie(config)# interface gigabitEthernet 0/0
Ruijie(config-if)# ip inspect 123 in
Ruijie(config)# show ip inspect all
Inspection Rule Configuration
Inspection name abc
ftp
mms
Inspection name 123
mms
h323
Interface Configurationn
Interface gigabitEthernet 0/0
Inbound inspection rule is 123
mms
h323
```

Understanding TCP Sequence Number Tracking

Overview

The purpose of TCP sequence number check is to guard against intrusion such as TCP session hijacking. It determines whether a data packet is valid by recording and tracking send sequence numbers, acknowledgement sequence numbers, and receive windows of both sides of a TCP connection.

Configuring TCP Sequence Number Tracking

Enabling or Disabling the TCP Sequence Number Tracking Function

The TCP sequence number tracking function is disabled by default. To enable this function, run the **ip inspect** command in interface configuration mode; or you may use **no** form of this command to disable the function.

Configuring TCP Sequence Number Tracking Rules

The TCP sequence number tracking function configuration basically agrees with the special protocol with the difference existing in that TCP protocol is added to **ip inspect name** rule library, e.g.,.

```
Ruijie(config)# ip inspect name abc tcp
```

The preceding command indicates that TCP is added to the detection rule named "abc".

Then, it is still necessary to apply the configured ip inspect name detection rule to the interface, e.g.,.

```
Ruijie(config)# interface gigabitEthernet 0/0
Ruijie(config-if)# ip inspect abc in
```

The preceding command applies the detection rule named "abc" in the inbound direction of gigabit 0/0 interface.

Understanding Session Limit

Overview

This function is mainly designed to prevent flow flood attacks generated at certain IP address or IP network segment by limiting the speed of session establishment and total number of concurrent sessions.

Configuring a Session Limit

Before configuring the session limit function, users need to configure an ACL rule first to define the scope of the session limit function, e.g.,.

If users want to limit all sessions through ports whose sources and destination IP addresses are specified in the limit function, configure the following ACL first.

```
access-list 1 permit any
```

Then configure the rule in interface mode.

For example, now users want to configure a rule under gigabit 0/0 interface, and the rule takes effect in the inbound direction of this interface, the limit scope shall be ACL-defined scope, the speed of session establishment shall be 100 per second and concurrent sessions shall be 100,000 in total. Sessions that meet such requirements shall be allowed while others are blocked and the blocked session will be recorded in a log.

```
Ruijie(config)# in gi 0/0
Ruijie(config-if)# session-limit access-group 1 rate 100 concurrent 100000 in log
```

Use the **no** form of this command to delete configuration.

Viewing Session Limit Information

show session-limit config

```
Ruijie(config-if)# show session-limit config
============[ Show gigabitEthernet 0/0's config]==============
Input
session-limit access-group 1 rate 12 concurrent 123 in log
session-limit access-group 12 rate 20 concurrent 100 in log
session-limit access-group 13 rate 20 concurrent 100 in log
session-limit access-group 14 rate 20 concurrent 100 in log
------------------------------------------------------------------------------------------
-----------------
 Output
session-limit access-group 1 rate 12 concurrent 123 out log
===========[ Show gigabitEthernet 0/0's config end]=============
```

show session-limit del-rule

With this function, you may find the commands used to view and delete configuration.

```
Ruijie(config-if)# show session-limit del-rule
```

```
======[ Show Cmd to del the rule on the gigabitEthernet 0/0 ]=====
Input
no session-limit access-group 1 rate 12 concurrent 123
in log
no session-limit access-group 12 rate 20 concurrent 100
in log
no session-limit access-group 13 rate 20 concurrent 100
in log
no session-limit access-group 14 rate 20 concurrent 100
in log
-------------------------------------------------------------------------------------------
-----------------
Output
no session-limit access-group 1 rate 12 concurrent 123
out log
==[ Show Cmd to del the rule on the gigabitEthernet 0/0's end ]=
```

show session-limit statistics

```
Ruijie(config-if)# show session-limit statistics
=========[ Show gigabitEthernet 0/0's Statistics ]========
Input
matches access-group  : 1
[Configure]:  new_session_rate : 12 , concurren : 123
[Statistics]:  conformed 2247 sessions, blocked 0 sessions
matches access-group  : 12
[Configure]:  new_session_rate : 20 , concurren : 100
[Statistics]:  conformed 0 sessions, blocked 0 sessions
matches access-group  : 13
[Configure]:  new_session_rate : 20 , concurren : 100
[Statistics]:  conformed 0 sessions, blocked 0 sessions
matches access-group  : 14
[Configure]:  new_session_rate : 20 , concurren : 100
[Statistics]:  conformed 0 sessions, blocked 0 sessions
-------------------------------------------------------------------------------------------
-----------------
Output
matches access-group  : 1
[Configure]:  new_session_rate : 12 , concurren : 123
[Statistics]:  conformed 0 sessions, blocked 0 sessions
=========[ Show gigabitEthernet 0/0's Statistics End ]=======
```

Understanding Flow Management

Overview

The purpose of flow management is to prevent some users or applications from occupying excessive resources (e.g., bandwidth). Besides, flow limit is a simple and direct way to guard against ICMP flood and UDP flood attacks when other means of defense are void.

Configuring Flow Management

During flow management configuration, ACLs are used to control bandwidth quota of users, maximum number of concurrent connections, and number of new connections. Bandwidth is classified into uplink bandwidth and downlink bandwidth. If same bandwidth is specified on the uplink and downlink, the system automatically changes the keyword to **both**. The number of concurrent connections and speed of connection establishment are optional and can be left unspecified.

To configure this function, run the **ip rate-control** command in interface configuration mode.

You can use the **no** form of this command to disable this function.

It should be noticed that the command is effective only on the outbound interface.

Understanding Others

Enabling Session Log

At the end of a session, some information of the session, including source IP address, destination IP address, protocol, port, bytes sent and received, session duration, may need to be sent to the log server for future analysis. The session log function is disabled by default. To enable this function, run the **ip session log-on** command.

Sometimes a large number of session logs may be generated. In this case, some session logs may be lost due to limits on network transmission and processing capability of the log server.

Confiuguring Session Timeout

One session that remains inactive in a certain time will be considered completed. This period of time is called the timeout time of a session. Session timeout varies with session statuses. The system has different timeout settings for sessions in different statuses, which does not need to be changed in normal conditions. If you want to change timeout settings, run the **ip session timeout** command.

Configuring Abnormal Session Status Restriction

As for some abnormal session status, it is conducive to security enhancement to control the number of the packets sent from the source terminal. Generally, an appropriate threshold of different abnormal session status has been configured by the system and needs no change. When a change is needed, run the **ip session threshold** command.

Enabling Strict Status Tracking

Strict status tracking applies to TCP connection establishment and ICMP error packets. It interrupts connections in case of abnormal TCP connection (e.g., non-SYN packet) and reception of unreachable ICMP packets. Misreport may occur, therefore, strict status tracking is disabled by default. To enable this function, run the **ip session track-state-strictly** command. It is recommended that you enable this function when there is a high security requirement for better attack defense capability and disable it in internal networks or private networks to protect key services.

Note

When running the **ip session track-state-strictly** command to start the FW module for strict status tracking, the system will interrupt established TCP connections in order to trigger stream creation for effective tracking of the status. As a result, some established TCP services like telnet, ftp, etc will be interrupted when the command is used. Be cautious when using this command.

Enabling ICMP Reverse Flow Check

When ICMP reverse flow check is disabled, only the life time of ICMP reverse flows will be refreshed and flows are deleted and re-established by themselves in case of a ping failure. When the function is enabled, life time of both forward and reverse flows will be refreshed. This function is disabled by default. If there is a routing imbalance, packets are sent only in one direction and packets may be lost in a ping test. In this case, you can enable this function to avoid packet loss.

Configuring Connection Filtering

Invalid IP packets affect the transmission of normal packets. To prevent communication or attacks that hinge on invalid packets, you can enable connection filtering.

Before enabling this function, you need to know characteristics of invalid packets such as the source IP address, destination IP address, protocol, and port and then configure an ACL rule to define the range of forbidden packets. Finally, you can run the **ip session filter** *acl_id* command.

This function takes effect globally. It checks and filters forward and reverse flows, and discards packets that meet the filter rule without creating flow entries on the platform.

The connection filtering function is not disabled or displayed by default.

Network Security Protocol (IPSec)

Overview of IPSec

Purposes of Encryption

Data transmitted on a network without any protection measures is vulnerable to various kinds of attacks. When the data passes a device, any one who accesses this device can read, tamper or forge the data. For example, the protocol analyzer (such as sniffer) can be used to read packets and obtain confidential information. Inside an organization, the malicious users can tamper packets and perform destructive activities by interfering, reducing or blocking network traffic. Hence, it is extremely important to encrypt private, confidential and emergent data when it is transmitted.

Ruijie Networks products support the IPSec and IKE protocols, ensuring that the data is transmitted securely in a network without any protection measures.

- The IPSec protocol is an open standard framework developed by IETF. It works in the network layer to provide encryption and authentication for the traffic between the devices that provide the IPSec protocol services. IPSec can protect all or part of the data above the IP layer. It provides the following optional security services: data confidentiality, data integrity, data origin authentication, and anti-replay. These functions prevent the data from being monitored, tampered and forged when being transmitted over the network.
- IKE is a key management protocol standard that should be used with IPSec. IKE works in the UDP layer to provide secure key exchange and management mechanism. Since IPSec can be used independently, IKE will make IPSec more flexible and easy to configure, strengthening the security.

Supported Standards

Ruijie Networks products implement the following encryption standards:

- IPSec: It specifies a set of security architectures and provides data confidentiality, integrity and data authentication services between IPSec entities. It can protect one or more data streams between hosts, between subnets, and between security gateways.
- AH: It provides the data authentication service and the anti-replay service.
- ESP: It provides the data encryption service, the optional data authentication service, and the anti-replay capability.
- DES: It is an encryption algorithm that uses a 64-bit key to encrypt the packet (there are 56 significant bits).
- 3DES: It is an encryption algorithm that uses a 192-bit key to encrypt packets (there are 168 significant bits).
- AES: It is a sub-key, 128-bit data input algorithm, key length is 128. As the next generation data encryption standard, AES boasts high security, high performance, high efficiency, easy-to-use and flexibility
- NULL: It is the null encryption algorithm that encapsulates, instead of encrypts, packets.
- MD5-HMAC: (Message Digest 5) It is a HASH algorithm used to verify packets and prevent them from being modified.
- SHA-HMAC: (secure HASH algorithm) It is a HASH algorithm used to verify packets and prevent them from being modified.

- IKE: This protocol implements the Oakley and Skeme key exchange protocols within the ISAKMP (Internet Security Association and Key Management Protocol) framework. It performs IPSec end-point authentication, IPSec parameter negotiation and key exchange.
- ISAKMP: It defines the format and parameters of the payload in data exchange, and the key negotiation mode.
- Diffie-Hellman: It is a public key encryption protocol that allows both parties involved in the exchange to establish shared secrecy on an insecure channel.

Terms

Anti-replay: It is a security service that allows recipients to deny outdated packets or packet copies to avoid being attacked. It is a security association that is used for IKE negotiation and provides the authentication service.

Note

The manually created security association does not support the anti-replay function. Only the security associations that pass IKE negotiation support anti-replay.

Data authentication: It includes the following two concepts:

- Data integrity: Check whether the data has been modified.
- Data origin authentication: Check whether the data is really sent by the declared sender.

Data confidentiality: It protects the data from being snooped.

Data stream: It refers to the specific communication data that has a source address/mask, destination address/mask, the next protocol field of IP, and source and destination port IDs. The protocol and port fields can be specified using "any". All the traffic of a certain association that meets the above conditions is called a data stream. A data stream may represent a TCP connection between two hosts, or all the traffic between two subnets.

Peer: It refers to the device involved in IPSec or other devices.

Security Association (SA): It refers to a logical connection that provides the security service for a specific data stream. This security service has such parameters as specific security protocol, security algorithm, key, and data stream description. There are two types of security association: IPSec and IKE. The IPSec SA provides the IPSec protection function for data and allows users to establish a connection either manually or through IKE negotiation. The IKE SA is used to protect the negotiation data of IKE.

Security Parameter Index (SPI): SPI is a 32-bit integer, which is combined with a destination IP address and a security protocol type to form the unique ID of a SA. When a SA is established by using IKE, the SPI value of each SA is a pseudo-random inherited digit. If IKE is not used, specify an SPI value for each SA.

Security association lifetime: It refers to the validity period of a SA. The manually established security association has no lifetime, that is, it can be used permanently until a user deletes it manually. The lifetime of the SA established through IKE negotiated is negotiated with the remote IKE entity. The SA will be deleted once its lifetime expires, and IKE will negotiate about a new SA.

Transform set: The transform set describes the security suite that consists of a security protocol (AH or ESP) and an algorithm. For example, a transform set defines use of the ESP protocol and the DES encryption algorithm.

Crypto map entry: The crypto map entry associates the transform set with the data stream, and describes the peer address, and parameters necessary for communication. It fully describes the contents necessary for IPSec communication with the remote peer. An IPSec SA can be established only by using the crypto map entry.

At present, IPSec can be used to send IP packets in unicast manner only. Because the IPSec workgroup has not released the group key, now IPSec does not support IP packet multicast or broadcast.

If a device uses NAT, the static NAT should be configured, so that IPSec can work normally. NAT must be performed before IPSec encapsulation of the device, that is, IPSec should use the IP address of the public network.

IPSec Configuration

Overview of IPSec Working Process

IPSec provides a secure channel for two IPSec peers, such as two devices. You can define which sensitive data streams should be protected. These data streams will be transmitted along the secure channels. Moreover, you can define parameters to protect these sensitive packets by specifying parameters for these channels. When IPSec detects such a sensitive packet, it will establish a secure channel, through which this packet is sent to the remote peer.

The sensitive data streams can be defined by configuring the access list. Describe the sensitive data streams to be protected on the basis of the source/destination address, protocol and port in the access list. After configuring the access lists, use a crypto map set to apply these access lists to the interface, so that the interface protects the specific incoming and outgoing data streams.

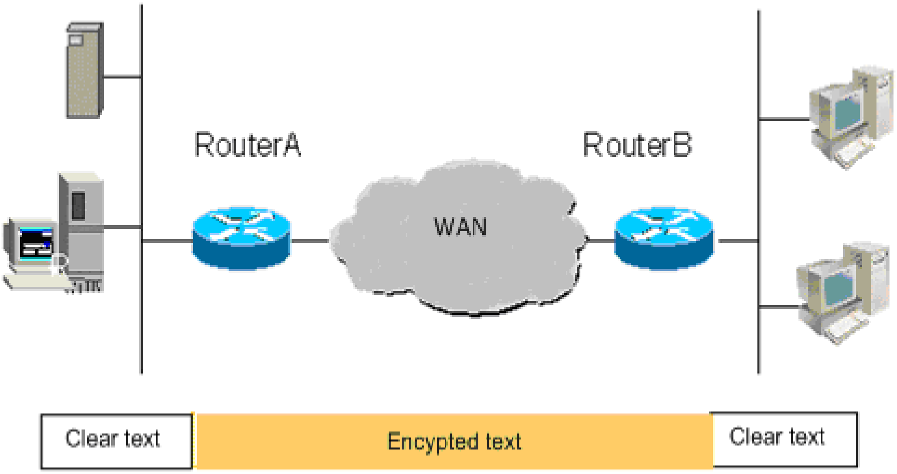
One crypto map set can have multiple entries, each of which corresponds to a different access list. The device finds the entry that matches the current traffic by sequence (the device tries to match the packet with the access list specified by the entry). When a packet matches a permit entry in the specific access list, if the crypto map entry is labeled as ipsec-manual, IPSec is triggered directly to process the data stream securely; if the crypto map entry is labeled as ipsec-isakmp, when an IPSec SA has been established, IPSec protection is provided to the data directly, otherwise IKE negotiation will be triggered automatically to create an IPSec SA. If the user does not configure IPSec or IKE parameters properly, it will be impossible to establish a SA, and the packets will be discarded.

Once a SA is established, the outgoing packet will be encrypted by IPSec and authentication information is filled in before it is sent to the peer. This packet is an incoming packet of the peer, which finds the related SA, and decrypts, authenticates and restores the packet.

The crypto map entry also specifies a transform set that defines the combination of the algorithm and protocol mode used by IPSec. Two IPSec peers must finally use the same transform set in order to communicate effectively.

The following figure shows an example of implementing IPSec protection between subnets:

Figure 7 Implementation of IPSec protection between subnets



IPSec Configuration Tasks

The ultimate purpose of IPSec configuration task is to establish an IPSec SA. An IPSec SA can be established manually or through negotiation by IKE. Manual configuration does not need IKE, but requires more parameters to be specified and has lower level of security. To establish a SA through IKE negotiation, you need to configure the IKE parameters besides configuring the IPSec parameters, so it has a higher level of security.

IPSec configuration tasks include:

- **Configure the default lifetime (optional)**: This is an optional step. You can use this command to modify the default lifetime value of the system. If there is no special description, IKE will use this lifetime value for negotiation, so that the lifetime of IPSec is not longer than the default lifetime.
- **Create an encryption access list**: An encryption access list determines the data streams to be protected. IPSec needs to rely on the encryption access list to filter incoming/outgoing packets. It provides IPSec protection for the matched outgoing data, and checks the validity of the matched incoming packets.
- **Define a transform set**: A transform set describes how to protect data streams. The transform set is a combination of the specific security protocol and algorithm. It specifies an algorithm, a security protocol, and a data encapsulation mode. To specify the degree of and requirements for protection of the data, the user must define an appropriate transform set here in advance.
- **Create a crypto map entry**: To create a crypto map entry, associate the predefined access list with the transform set, and define the key and the peer address to form a complete IPSec scheme description.
- **Configure a multicast policy:** Disable IPSec encapsulation of multicast and broadcast packets.
- **Apply a crypto map entry to an interface**: This action activates IPSec scheme defining. It applies a crypto map entry to an interface to make the crypto map set start working on the interface.
- **Create a Profile crypto map entry:** Define IPSec encryption policies of dynamic multipoint VPN (DMVPN).
- **Apply a Profile crypto map entry to a tunnel interface**: Activate the IPSec functions of DMVPN.
- **Configure extended authentication mode:** This action is used for extended authentication.
- **Configure IPSec packet filter:** Decapsulated packets are no longer filtered.
- **Configure IPSec MIB:** It sends IPSec monitoring information to the SNMP server. This function is disabled by default and needs to be enabled using a command.
-

- **Monitor and maintain IPSec**: Monitor and maintain IPSec, view and adjust the IPSec parameters, and judge whether IPSec works normally.

Note

IKE uses UDP port 500. The IPSec ESP and AH protocols are numbered 50 and 51 respectively. If access list (firewall) filtering data has been configured on the device, then before configuring IPSec, please make sure the traffic for protocols 50 and 51 and UDP port 500 on the interface used by IPSec is not blocked. If possible, add a statement to the access list to explicitly allow the traffic.

Configuring Default Lifetime

To configure the default lifetime, run the following command in global configuration mode or privileged user configuration mode:

Command	**Function**
Ruijie(config)# **crypto ipsec security-association lifetime seconds** *seconds*	Changes the global lifetime limit of IPSec SA. This command will cause the SA timeout after the specified seconds elapse.
Ruijie(config)# **crypto ipsec security-association lifetime kilobytes** *kilobytes*	Changes the global traffic lifetime of IPSec SA. This command will cause the SA timeout after the transmitted traffic (in KBs) protected by IPSec using this SA reaches a specified value.
Ruijie# **clear crypto sa** or Ruijie# **clear crypto sa peer** {*ip-address* | *peer-name*} or Ruijie# **clear crypto sa map** *map-name*	Clears an existing SA. This will immediately interrupt all the existing SAs. The subsequent SAs will use a new lifetime. Otherwise, all the existing SAs will expire on the basis of the original lifetime.

Note

Use the **clear crypto sa** command without parameters to clear the entire SA database. This will also clear the active encryption processes. You can use such keywords as peer and map to clear only one subnet from the SA database. For detailed information, refer to the command reference for **clear crypto sa**.

The default lifetime of the system is 1-hour communication (3600 seconds) or 4,608,000KB traffic (continuous communication for 1 hour at the rate of 10 MBit/s). If the user accepts the default value, skip this step. This default lifetime is used if there is no special description in the crypto map entry. When negotiating the lifetime of IPSec, IKE uses the smaller value of those of the local end and the peer. When the lifetime of the IPSec SA expires, IKE will negotiate again and replace a new set of parameters and key for IPSec to make it start working again.

The SA (and the related key) is timeout on the basis of the lifetime that expires earliest: Use the seconds (specified by the keyword seconds) or the kilobytes of transmitted traffic (specified by the keyword kilobytes). The manually established SA (established by the crypto map entry identified as ipsec-manual) has no lifetime limit.

In order to make sure that a new SA is available when the original SA expires, the new SA must be negotiated before the original SA expires. When there are 30 seconds left before the lifetime expires, or when there are 256 Kbytes left before the traffic that passes this channel reaches the lifetime (determined by the peer that reaches the lifetime first), a new SA is negotiated.

If no traffic takes this channel throughout the lifetime of a SA, this SA will be released but negotiation of a new SA will not be underway when the lifetime elapses. In this case, a new SA will be negotiated only when IPSec finds another packet that should be protected.

Configuring Automatic Disconnection for Idle Tunnels

To configure automatic disconnection for global idle tunnels, run the following command in global configuration mode or privileged EXEC mode:

Command	Function
Ruijie(config)#**crypto ipsec security-association idle-time** *sec*	Specifies the disconnection period for global idle tunnels in seconds. The value range is 60–86400.

DF bit Override Function of IPSec Tunnel

The DF bit override function allows users to specify whether the device is reset, is set to 1, or copies the encapsulated header.

The DF bit on the IP header determines whether the device can fragment the packet. Value 1 indicates that this packet cannot be fragmented, and value 0 indicates that the packet can be fragmented. This function in IPsec tunnel mode allows the device to control whether the DF bit of the packet IP header encapsulated by IPSec is determined by the DF bit value of the original IP header. Only tunnel mode supports this feature.

To configure the DF bit value for all the interfaces, run the following command in global configuration mode:

Command	Function
Ruijie(config)# **crypto ipsec df-bit** [**clear** | **set** | **copy**]	Sets the DF bit of the IP external header for all the interfaces in tunnel mode.

In the following example, the device is configured to clear the DF bit globally, and copy the DF bit on FastEthernet0/0. This way, all the interfaces other than FastEthernet0/0 allow packets larger than the MTU size to be sent (in fragments), while FastEthernet0/0 must determine whether to allow the device to fragment the packet according to the DF bit in the original IP header.

```
crypto isakmp policy 1
hash md5
authentication pre-share
crypto isakmp key 0 DELaware address 192.168.10.66
crypto isakmp key 0 Key-What-Key address 192.168.11.19
!
!
crypto ipsec transform-set BearMama ah-md5-hmac esp-des
crypto ipsec df-bit clear
!
!
```

```
crypto map armadillo 1 ipsec-isakmp
set peer 192.168.10.66
set transform-set BearMama
match address 101
!
crypto map basilisk 1 ipsec-isakmp
set peer 192.168.11.19
set transform-set BearMama
match address 102
!
!
interface FastEthernet0/0
ip address 192.168.10.38 255.255.255.0
ip broadcast-address 0.0.0.0
crypto map armadillo
crypto ipsec df-bit copy
!
interface FastEthernet0/1
ip address 192.168.11.75 255.255.255.0
ip broadcast-address 0.0.0.0
crypto map basilisk
!
```

Creating Encryption Access Lists

The encryption access list is used to define which data streams should be encrypted, and which should not be encrypted. For example, you can create an encryption access list to protect all the IP traffic between Subnet A (192.168.202.0/24) and Subnet B (192.168.12.0/24) (access list 120), or the IP traffic between Host A and Host B (access list 101):

```
access-list 120 permit ip 192.168.12.0 0.0.0.255 192.168.202.0 0.0.0.255
access-list 101 permit ip host 2.2.2.2 host 2.2.2.1
```

The encryption access list specified by the IPSec crypto map entry has the following four major functions:

- Filter the outbound traffic that is encrypted by using IPSec (permit = protect).
- When starting to negotiate an IPSec SA, indicate which data streams are protected by the new SA (indicated by a single permit entry).
- Process the inbound traffic, so as to filter and discard those traffic that should have been protected by IPSec.
- When handling the IKE negotiation initiated by the IPSec peer, determine whether to accept the IPSec SA request that represents the requested data stream (only the crypto map entry ipsec-isakmp should be negotiated). You must make sure that the access lists of peers at both ends match. It is recommended that the access lists of peers at both ends are consistent.

To configure the encryption access list, run the following command in global configuration mode:

Command	Function
Ruijie(config)# **access-list** *access-list-number* {**deny** | **permit**} *protocol source source-wildcard destination destination-wildcard* [**log**] Or: Ruijie(config)# **ipv6 access-list** ipv6-acl-name Ruijie(config-ipv6-acl)# **{deny | permit}** *protocol source source-wildcard destination destination-wildcard*	Describes the data stream in terms of its source/destination address and its wildcard, communication protocol and communication port. If the keyword **permit** is used, the policy described in the related crypto map entry will provide encryption protection for all the IP traffics that meet the specified conditions. The keyword **deny** can be used to prevent the traffic from being encrypted by the specific crypto map entry.
Ruijie(config-exp-nacl)# **exit**	Exits ACL configuration mode.

If the keyword **permit** is used, the policy described in the related crypto map entry will provide encryption protection for all the IP traffic that meets the specified conditions. The keyword **deny** can be used to prevent the traffic from being encrypted by the specific crypto map entry.

Note

It is recommended that you define a mirrored encryption access list on the remote peer for each encryption access list defined on the local peer. Otherwise, some data is not protected or the SA cannot be established. Since the ACL has priority, the inclusion relation of ACEs should be noted during the configuration. In case of conflict, the ACE having a higher priority takes effect.

The keyword **any** should be used with great care because it will discard lots of broadcast information and make the device unable to work normally. The encryption access list is not specially designed for and used by IPSec. IPSec uses the extended IP access list, so the value of **access-list-number** ranges from 100 to 199]. If no port is defined, this encryption access list can be used for the data stream in either the inbound direction or the outbound direction.

For example, when you want to protect the IP traffic between Subnet A (192.168.12.0/24) and Subnet B (192.168.10.0/24), the following access list should be defined for the device:

```
access-list 101 permit ip 192.168.12.0 0.0.0.255 192.168.10.0 0.0.0.255
```

For example, if you want to protect the TCP traffic between Subnet A (192.168.12.0/24) and Host C (202.101.11.3), the following access list should be defined for the device:

```
access-list 120 permit tcp 192.168.12.0 0.0.0.255 202.101.11.3 0.0.0.0
```

If port filtering is defined, the destination address in the encryption access list provides the service for this port.

For example, if you need to protect the Telnet traffic between Host D (1.1.1.1) and Host E (2.2.2.2) that provides the Telnet service, define as follows on the device:

```
access-list 133 permit tcp 1.1.1.1 0.0.0.0 2.2.2.2 0.0.0.0 eq telnet
```

Caution

In terms of permit ipv6 any any encryption access list, Ruijie devices are not compatiable with Cisco devices. Because Cisco devices will encrypted “neighbor request packet” and “neighbor advertisement packet” (Similar to ARP packets of IPv4), the IPSec data communication between two non-direct devices failed. To avoid the preceding issue, Ruijie devices will not encrypt “neighbor request packet” and “neighbor advertisement packet”.

Defining Transform Set

Transform set is a combination of the specific security protocol and algorithm. During negotiation of the IPSec SA, the peer must use the same specific transform set to protect the specific data stream.

Because there is no anti-replay negotiation process between peers for the manually established SA, the same transform set must be specified for the two peers. Change to the definition of the transform set will apply to negotiation of the subsequently established SA, instead of the existing SA. If you want these new settings to take effect immediately, use the **clear crypto sa** command to clear all or part of the SA database.

To define a transform set, run the following command in global configuration mode:

Command	**Function**
Ruijie(config)# **crypto ipsec transform-set transform-set-name** *transform1* [*transform2* [*transform3*]]	The transform parameter is an algorithm supported by the system. Algorithms can be combined according to a certain rule.
Ruijie(cfg-crypto-trans)# **mode** {**tunnel** | **transport**} (Optional)	Changes the mode associated with the transform set. Mode setting is useful only for the communication where both the source and the destination addresses and those of the IPSec peer, while not useful for other communications (all other communications are performed in the tunnel mode).
exit	Exits crypto transform configuration mode.
Ruijie# **clear crypto sa** or Ruijie# **clear crypto sa peer** {*ip-address* | *peer-name*} or Ruijie# **clear crypto sa map** *map-name*	Clears the existing SAs, so as to make sure that any change to the transform set applies to the subsequently established SAs (the manually established SAs will be reestablished immediately)

Present below are all the transform sets supported by the system:

transform1 [transform2]	**Description**
ah-md5-hmac	AH protocol and MD5 HMAC algorithm
ah-sha-hmac	AH protocol and SHA HMAC algorithm
ah-sm3-hmac	AH protocol and SM3 HMAC algorithm
esp-des	ESP protocol and DES encryption algorithm
esp-3des	ESP protocol and 3DES encryption algorithm
esp-aes-128	ESP protocol and aes encryption algorithm with key length being 128.
esp-aes-192	ESP protocol and aes encryption algorithm with key length being 192.

transform1 [transform2]	Description
esp-aes-256	ESP protocol and aes encryption algorithm with key length being 256.
esp-sm4	ESP protocol and SM4 encryption algorithm with key length being 128.
ah-md5-hmac esp-des	The AH protocol is located outside, and the MD5 HMAC authentication algorithm is used. The ESP protocol is located inside, and the DES encryption algorithm is used
ah-sha-hmac esp-des	The AH protocol is located outside, and the SHA HMAC authentication algorithm is used. The ESP protocol is located inside, and the DES encryption algorithm is used
ah-md5-hmac esp-des esp-md5-hmac	The AH protocol is located outside, and the MD5 HMAC authentication algorithm is used. The ESP protocol is located inside, and the DES encryption algorithm and the MD5 HMAC authentication algorithm are used
ah-md5-hmac esp-null esp-md5-hmac	The AH protocol is located outside, and the MD5 HMAC authentication algorithm is used. The ESP protocol is located inside, and the null encryption algorithm and the MD5 HMAC authentication algorithm are used
ah-md5-hmac esp-des esp-sha-hmac	The AH protocol is located outside, and the MD5 HMAC authentication algorithm is used. The ESP protocol is located inside, and the DES encryption algorithm and the SHA HMAC authentication algorithm are used
ah-md5-hmac esp-null esp-sha-hmac	The AH protocol is located outside, and the MD5 HMAC authentication algorithm is used. The ESP protocol is located inside, and the null encryption algorithm and the SHA HMAC authentication algorithm are used
ah-sha-hmac esp-des esp-md5-hmac	The AH protocol is located outside, and the SHA HMAC authentication algorithm is used. The ESP protocol is located inside, and the DES encryption algorithm and the MD5 HMAC authentication algorithm are used
ah-sha-hmac esp-null esp-md5-hmac	The AH protocol is located outside, and the SHA HMAC authentication algorithm is used. The ESP protocol is located inside, and the null encryption algorithm and the MD5 HMAC authentication algorithm are used
ah-sha-hmac esp-des esp-sha-hmac	The AH protocol is located outside, and the SHA HMAC authentication algorithm is used. The ESP protocol is located inside, and the DES encryption algorithm and the SHA HMAC authentication algorithm are used
ah-sha-hmac esp-null sp-sha-hmac	The AH protocol is located outside, and the SHA HMAC authentication algorithm is used. The ESP protocol is located inside, and the null encryption algorithm and the SHA HMAC authentication algorithm are used

transform1 [transform2]	Description
esp-des esp-md5-hmac	For the ESP protocol, the DES encryption algorithm and the MD5 HMAC authentication algorithm are used.
esp-null esp-md5-hmac	For the ESP protocol, the null encryption algorithm and the MD5 HMAC authentication algorithm are used.
esp-des esp-sha-hmac	For the ESP protocol, the DES encryption algorithm and the SHA HMAC authentication algorithm are used.
esp-null esp-sha-hmac	For the ESP protocol, the null encryption algorithm and the SHA HMAC authentication algorithm are used.
esp-3des	ESP protocol and 3DES encryption algorithm
esp-3des esp-sha	For the ESP protocol, the 3DES encryption algorithm and the SHA HMAC authentication algorithm are used.
esp-3des esp-md5	For the ESP protocol, the 3DES encryption algorithm and the MD5 HMAC authentication algorithm are used.
ah-md5-hmac esp-des	The AH protocol is located outside, and the MD5 HMAC authentication algorithm is used. The ESP protocol is located inside, and the 3DES encryption algorithm is used
ah-sha-hmac esp-des	The AH protocol is located outside, and the SHA HMAC authentication algorithm is used. The ESP protocol is located inside, and the 3DES encryption algorithm is used
ah-md5-hmac esp-3des esp-sha	The AH protocol is located outside, and the MD5 HMAC authentication algorithm is used. The ESP protocol is located inside, and the 3DES encryption algorithm and the SHA HMAC authentication algorithm are used
ah-sha-hmac esp-3des esp-sha	The AH protocol is located outside, and the SHA HMAC authentication algorithm is used. The ESP protocol is located inside, and the 3DES encryption algorithm and the SHA HMAC authentication algorithm are used
ah-md5-hmac esp-3des esp-md5	The AH protocol is located outside, and the MD5 HMAC authentication algorithm is used. The ESP protocol is located inside, and the 3DES encryption algorithm and the MD5 HMAC authentication algorithm are used
ah-sha-hmac esp-3des esp-md5	The AH protocol is located outside, and the SHA HMAC authentication algorithm is used. The ESP protocol is located inside, and the 3DES encryption algorithm and the MD5 HMAC authentication algorithm are used

Note

Generally, the combination esp-des (without data authentication) will satisfy your requirement. If you want to verify data, you can choose esp-des esp-md5-hmac or esp-des esp-sha-hmac.

Configuring IPSec MIB

IPSec MIB management involves statistics of data streams and encrypted/decrypted data packets and it may affect performance of IPSec data communication in some certain. Therefore, the MIB statistical function is disabled by default. To access the MIB node of IPSec, you need to enable the IPSec MIB function using the CLI command.

Command	**Function**
Ruijie(config)# **crypto mib enable**	Configures IPSec MIB statistics function.

Configuring Multicast Policies

If an ACL covers multicast and broadcast addresses, IPSec encapsulation will be applied to packets related to these addresses. To skip IPSec encapsulation, run the following command:

Command	**Function**
Ruijie(config)#**crypto ipsec multcast disable**	Disables encapsulation of multicast and broadcast packets.

Creating Crypto Map Entry

The crypto map entry can be configured in the following aspects:

- Which traffic should be protected by IPSec: Associate the configured encryption ACL.
- Where the traffic protected by IPSec will be sent to: Which is the remote IPSec peer.
- Local address used for IPSec communication: Apply the crypto map set to the interface. IPSec uses the address of the communication interface as the address of the local peer.
- Which IPSec security policies should be applied to traffic: Choose from the list that consists of one or more transform sets.
- Lifetime of the SA.
- Whether the SA is established manually or through IKE negotiation.

The crypto map entries that have the same crypto map name (but with different map sequence numbers) constitute a crypto map set. Apply the crypto map set to the interface, so that all the IP traffic that passes this interface is judged according to the crypto map set applied to the interface. If a crypto map entry finds an outbound IP channel that should be protected, and the crypto map specifies use of IKE, the SA will be negotiated with the remote peer according to the parameters in this crypto map entry. If the crypto map entry specifies use of the manually established SA, then a SA must have been established during configuration. The data is encrypted for transmission once the SA is established successfully either manually or through IKE negotiation. If negotiation of SA fails, the data is discarded.

The policy described in the crypto map entry will be used during negotiation of SA. To carry out IPSec smoothly between two IPSec peers, the crypto map entries of the two peers must include mutually compatible configuration statements. When two peers try to establish a SA, both of them must have at least one crypto map entry that is compatible with the crypto map entry of the remote peer and at least meets the following conditions:

- The crypto map entry must include a compatible encryption access list (such as mirrored map access list).
- The crypto map entries at both sides must identify the address of the peer (unless the peer is using a dynamic crypto map).
- The crypto map entries must have at least one identical transform set.

Only one crypto map set is applied to a single interface. The crypto map set contains IPSec/IKE or combination of IPSec/manual entry. If you create multiple crypto map entries for a given interface, you have to use the *seq- num* parameter of the map entry to sort these map entries again. The smaller the value of *seq-num*, the higher the priority.

Multiple crypto map entries must be created for a single interface if one of the following situations exists.

- If different data streams on this interface will be processed by different IPSec peers.
- If you want to apply different IPSec securities to different types of traffic (destined for the same or different peers). For example, you want that the traffic among the subnets in a group is authenticated, while the traffic among other subnets is both authenticated and encrypted. In this case, different types of traffic should be defined in two different access lists, and a separate crypto map entry must be created for each encryption access list.

Creating a SA manually

To create a SA manually, run the following commands in global configuration mode at the beginning:

Command	**Function**
Ruijie(config)# **crypto map** *map-name* *seq-num* **ipsec-manual**	Specifies the crypto map entry to be created or modified. When using this command, you will enter crypto map configuration mode.
Ruijie(config-crypto-map)# **match address** *access-list-id* *or* Ruijie(config-crypto-map)# **match ipv6** *ipv6-acl-name*	Specifies an access list for the crypto map list. This access list determines which traffic should be protected by IPSec, and which traffic should not be protected by the IPSec security defined in this crypto map entry.
Ruijie(config-crypto-map)# **match vrf** *vrf-name*	Specifies the crypto map list a VRF that are associated with the access list. Only when the packets under the VRF matching the access list can they be protected by IPSec.
Ruijie(config-crypto-map)# **set peer** {*hostname* | *ip-address*}	Specifies a remote IPSec peer. The traffic protected by IPSec will be sent to this peer. If IKE is not used, only one peer can be configured.
Ruijie(config-crypto-map)# **set transform-set** *transform-set-name*	Specifies which transform set to use. This transform set must be the same as the one specified in the corresponding crypto map entry of the remote peer. (If IKE is not used, only one transform set can be specified.)
Ruijie(config-crypto-map)# **set vrf** *vrf-name*	Specifies the VRF that are associated with tunnel.
Ruijie(config-crypto-map)# **set session-key inbound ah** *spi hex-key-data* or Ruijie(config-crypto-map)# **set session-key outbound ah** *spi hex-key-data*	If the specified transform set includes the AH protocol, you should use this command to set the AH Security Parameter Indexes (SPIs) and passwords for the protected outbound and inbound traffic. Here, the local inbound SPI, protocol and key must be the same as the outbound SPI, protocol and key of the remote peer, and vice versa.

Ruijie(config-crypto-map)# **set session-key inbound esp** *spi* **cipher** *hex-key-data* [**authenticator** *hex-key-data*] or Ruijie(config-crypto-map)# **set session-key outbound esp** *spi* **cipher** *hex-key-data* [**authenticator** *hex-key-data*]	If the specified transform set includes the ESP protocol, you should use this command to set the ESP security parameter indexes and passwords for the protected outbound and inbound traffics. If the transform set includes the ESP encryption algorithm, the encryption key must be provided. If the transform set includes the ESP authentication algorithm, the authentication key must be provided. Here, the local inbound SPI, protocol and key must be the same as the outbound SPI, protocol and key of the remote peer, and vice versa.
Ruijie(config-crypto-map)# **set mtu** *length*	Sets the side of a fragment in tunnel mode.
Ruijie(config-crypto-map)# **exit**	Exits the crypto map configuration mode and return to the global configuration mode.

Repeat the preceding steps to create other necessary crypto map entries.

The following shows a configuration example:

Local peer (router A) configuration:

# Define a transform set named myset

```
crypto ipsec transform-set myset esp-des
```

# Define a manual map set named mymap

```
crypto map mymap 3 ipsec-manual
 set peer 2.2.2.2
 set session-key inbound esp 301 cipher abcdef1234567890
 set session-key outbound esp 300 cipher abcdef1234567890
 set transform-set myset
 match address 101
!
access-list 101 permit ip 192.168.12.0 0.0.0.255 192.168.202.0 0.0.0.255
```

Remote peer (router B) configuration:

# Define a transform set named myset

```
crypto ipsec transform-set myset  esp-des
```

# Define a manual map set named mymap

```
crypto map mymap 3 ipsec-manual
 set peer 2.2.2.1
 set session-key inbound esp 300 cipher abcdef1234567890
 set session-key outbound esp 301 cipher abcdef1234567890
 set transform-set myset
 match address 101
 !
access-list 101 permit ip 192.168.202.0 0.0.0.255 192.168.12.0 0.0.0.255
```

Caution

The keyword **hex-key-data** is a hexadecimal number. The length of **hex-key-data** following the keyword cipher in the configuration command in the sixth step and the length of **hex-key-data** in the configuration command in the fifth step are determined by the encryption algorithm in use (at present, IPSec supports the DES, 3DES and AES encryption algorithms). The length of **hex-key-data** following the keyword authenticator in the configuration command in the sixth step is determined by the data authentication algorithms (including the SHA and MD5 algorithms) in use. In the above example, the encryption algorithm DES is used, and the length of 64 bits is required, so its value is set to abcdef1234567890 (equivalent to 0xabcdef1234567890). Because no data authentication algorithm is used in the above example, the key following authenticator is not configured. You can configure it simply by entering a string that contains digits from 1 to 9 and/or letters from a to f. It is unnecessary to identify it by 0x.

Table:

Name of Algorithm	Length of Key (bits)	Length of Entered Hexadecimal String (bytes)	Configuration Example
Des	64	8	Example: set session-key inbound esp 300 cipher abcdef1234567890
3Des	192	24	set session-key inbound esp 300 cipher abcdef1234567890 abcdef1234567890 abcdef1234567890
aes	128	16	set session-key inbound esp 300 cipher abcdef1234567890abcdef1234567890
Sm4	128	16	set session-key inbound esp 300 cipher abcdef1234567890abcdef1234567890
Md5	128	16	Example: set session-key inbound esp 302 cipher abcdef1234567890 authenticator abcdef1234567890abcdef1234567890
Sha	160	20	Example: set session-key inbound esp 302 cipher abcdef1234567890 authenticator abcdef1234567890abcdef1234567890abcd
Sm3	256	32	Example: set session-key inbound esp 302 cipher abcdef1234567890 authenticator abcdef1234567890abcdef1234567890abcdef1234567890abcdef1234567890

Caution

Generally, a full length of key should be configured. If the key is not complete, the device may append "0" (low security), or may not append "0" (this will cause negotiation of SA failed).

Use of the manual SA is the result prearranged by the local device and IPSec peer administrators. They may want to first use the manual SA for debugging, and then use the IKE-based SA or the remote peer does not support IKE.

Configuring Anti-Replay Window

The anti-replay window is the basic attack protection feature of IPSec. By default, when hash (MD5, SHA, etc) authentication mode is configured, the anti-reply feature will be enabled, yet you can still disable this feature through the following command:

Command	**Function**
Ruijie(config)#**crypto ipsec security-association replay disable**	Disables the anti-replay window.

Caution

Since QoS will divert traffic into different queues, thus leading to the disorder of packet transmission, and if IPSec enables anti-replay window in such a context, IPSec will drop all packets exceeding the window. Therefore, you can disable the anti-replay window to avoid packet loss, but it will also increase the possibility of being attacked.

Configuring Data Security Check

Data security check is the basic attack protection feature of IPSec. The criteria for attack judging is: if the packet which ought to be in encrypted text is received in plain text, such packet is considered unsafe and will be dropped. Under certain circumstances, data security check is not mandatory and can be disabled through the following command.

Command	**Function**
Ruijie(config)#**crypto ipsec optional**	Disables IPSEC data security check.

Caution

Data security check will result in significant resource overhead, and disabling this feature can save CPU resources. In the model of l2tp over ipsec, l2tp can force to enable IPSec, and thus only IPSec-encrypted packets are allowed. This feature can be used according to actual needs.

Configuring to use IKE to create a crypto map entry for the SA

To configure to use IKE to create a crypto map entry for the SA, run the following commands in global configuration mode in the beginning:

Command	**Function**
Ruijie(config)# **crypto map** *map-name seq-num* **ipsec-isakmp**	Specifies the crypto map entry to be created or modified. When using this command, you will enter crypto map configuration mode.

Command	**Function**
Ruijie(config-crypto-map)# **match address** *access-list-id* or Ruijie(config-crypto-map)# **match ipv6** *ipv6-acl-name*	Specifies an access list for the crypto map list. This access list determines which traffic should be protected by IPSec, and which traffic should not be protected by the IPSec security defined in this crypto map entry.
Ruijie(config-crypto-map)# **match vrf** *vrf-name*	Specifies the crypto map list a VRF that are associated with the access list. Only when the packets under the VRF matching the access list can they be protected by IPSec.
Ruijie(config-crypto-map)# **set peer** {*hostname* | *ip-address*}	Specifies a remote IPSec peer. The traffic protected by IPSec will be sent to this peer. You can configure multiple peers.
Ruijie(config-crypto-map)# **set local** *ip-address*	Sets the IP address for local negotiation. If no IP address is specified, the primary address of the interface is used.
Ruijie(config-crypto-map)# **set transform-set** *transform-set-name1* [*transform-set-name2…transform-set-name6*]	Specifies which transform set to use. List the transform set by priority (high priority first).
Ruijie(config-crypto-map)# **set security-association lifetime seconds** *seconds* or Ruijie(config-crypto-map)# **set security-association lifetime kilobytes** *kilobytes*	(Optional) Specifies a SA lifetime for the crypto map entry.
Ruijie(config-crypto-map)# **set security-association idle-time** *seconds*	(Optional) Specify the idle time timeout for the crypto map entry.
Ruijie(config-crypto-map)# **set exchange-mode** *main* | *aggressive*	Sets which mode is used to initiate negotiation using this static entry.
Ruijie(config-crypto-map)# **set pfs** *group1* | *group2*	Specifies the Diffie-Hellman group identification.
Ruijie(config-crypto-map)# **set mtu** *length*	Sets the fragment size in tunnel mode.
Ruijie(config-crypto-map)# **set vrf** *vrf-name*	Specifies the VRF that are associated with tunnel.
Ruijie(config-crypto-map)#**username** *name* **password {0|7}** *pass*	Configures the username and password used for extended authentication.
Ruijie(config-crypto-map)#**reverse-route** [**remote-peer** *ip-address*] [*distance*] [**tag** *tagvalue*] [**track** *trackvalue*] [**bfd**] [**weight** weightvalue]	Configures Ipv4 reverse routing
Ruijie(config-crypto-map)# **reverse-ipv6-route** [**remote-peer** *ip-address*] [*distance*] [**bfd**] [**weight** weightvalue]	Configures Ipv6 reverse routing
Ruijie(config-crypto-map)# **exit**	Exits crypto map configuration mode and returns to global configuration mode.

Repeat the preceding steps to create other necessary crypto map entries.

The following example shows how to configure to establish a SA using IKE:

Local peer (router A) configuration:

# Define a transform set named myset

```
crypto ipsec transform-set myset esp-des
```

# Define a map set named mymap that establishes a SA using IKE

```
crypto map mymap 3 ipsec-isakmp
 set peer 2.2.2.2
 set transform-set myset
 match address 101
!
access-list 101 permit ip 192.168.12.0 0.0.0.255 192.168.202.0 0.0.0.255
```

Remote peer (router B) configuration:

# Define a transform set named myset

```
crypto ipsec transform-set myset  esp-des
```

# Define a map set named mymap that establishes a SA using IKE

```
crypto map mymap 3 ipsec-isakmp
 set peer 2.2.2.1
 set transform-set myset
 match address 101
 !
access-list 101 permit ip 192.168.202.0 0.0.0.255 192.168.12.0 0.0.0.255
```

Note

RGOS supports both the manual SA and the IKE-established SA. The two modes of establishing SAs can also be added to the same crypto map set.

Caution

Use IKE to establish a SA because IKE will negotiate again and use a new key when the lifetime expires, ensuring data security. However, because the content encrypted by using the DES encryption algorithm may be cracked maliciously within a certain time, the key must be modified regularly if the SA is established manually.

Creating a dynamic crypto map

The dynamic crypto map (IKE is needed) requires less configuration. If the IP address of the remote peer is unknown during configuration, the dynamic crypto map function must be used. For example, a mobile subscriber is dynamically assigned an IP address. First, the mobile subscriber uses something other than the IP address, such as the domain name, for the local IKE authentication. Once the authentication is complete, the SA request that meets the dynamic crypto map can be processed, and this dynamic crypto map can accept requests that meet the local policy.

Understanding dynamic crypto map

Only IKE can use the dynamic crypto map. The dynamic crypto map entry acts as a policy template. The missing parameters can be obtained dynamically (IPSec negotiation) to meet requirement of the remote peer. It allows the remote peer and the device to exchange IPSec traffic even if the crypto map of the device does not fully satisfy the requirement of the remote peer.

The dynamic crypto map is used for the remote peer to initiate IPSec negotiation, not for the device to initiate new IPSec negotiation with the remote peer.

The dynamic crypto map set is referred to as a part of the crypto map. Any crypto map entry that refers to the dynamic map is the crypto map entry with the lowest priority in the crypto map set (namely it has the largest sequence number). This way, other crypto map entries will be evaluated first. The dynamic crypto map entry is checked when all the static crypto map entries do not match.

If the device accepts the request from the peer, it will create a new IPSec SA, and install a temporary crypto map entry, which is filled in with the negotiation result. At this point, the device uses a temporary crypto map entry as if it uses a normal crypto map entry. Once the SA expires, the temporary crypto map entry will be deleted.

For the static and dynamic crypto maps, if the incoming traffic not protected meets one permit statement in the access list, the traffic will be discarded because it is not protected by IPSec.

For the static crypto map entry, if the outgoing traffic meets the permit statement in the access list, and the corresponding SA has not been established, the device will initiate SA negotiation with the remote peer. For the dynamic crypto map entry, if a SA does not exist, the traffic is discarded directly (because the dynamic crypto map is not used to initiate a new SA negotiation).

Caution

The **any** keyword should be used carefully in the **permit** entry in the dynamic crypto map. Because **permit** may cover multicast and broadcast, **deny** must be used to exclude the broadcast and multicast traffic, and other traffic that is not protected by IPSec must also be excluded.

The dynamic crypto map entries are grouped into a set like the normal crypto map entries. A set contains crypto map entries that are grouped together using the same crypto map name but have different sequence numbers.

To create a dynamic crypto map entry, use the following commands in global configuration mode:

Command	Function
Ruijie(config)# **crypto dynamic-map** *dynamic-map-name dynamic-seq-num*	Creates a crypto map entry
Ruijie(config-crypto-map)# **set transform-set** *transform-set-name1* [*transform-set-name2...transform-set-name6*]	Specifies which transform set to use. List the transform set by priority (high priority first).

Ruijie(config-crypto-map)# **match address** *access-list-id* or Ruijie(config-crypto-map)# **match ipv6** *ipv6-acl-name*	(Optional) Specifies an access list for the crypto map list. This access list determines which traffic should be protected by IPSec, and which traffic should not be protected by the IPSec security defined in this crypto map entry. Note: Although the access list is optional to the dynamic crypto map, it is strongly recommended to configure it. If it is configured, the data stream ID suggested by the peer must match a permit entry in the crypto map access list. If it is not configured, the device accepts any data stream ID suggested by the peer.
Ruijie(config-crypto-map)# **match vrf** *vrf-name*	Specifies the crypto map list a VRF that are associated with the access list. Only when the packets under the VRF matching the access list can they be protected by IPSec.
Ruijie(config-crypto-map)# **set peer** {*hostname* | *ip-address*}	(Optional) Specifies a remote IPSec peer. You can configure multiple peers. This configuration is rare in the dynamic crypto map entry. The dynamic crypto map is often used when information about the peer is unknown.
Ruijie(config-crypto-map)# **set local** *ip-address*	Sets the IP address of the local peer. If no IP address is specified, the primary address of the interface is used.
Ruijie(config-crypto-map)# **set security-association lifetime seconds** *seconds* or Ruijie(config-crypto-map)# **set security-association lifetime kilobytes** *kilobytes*	(Optional) Specifies a SA lifetime for the crypto map entry.
Ruijie(config-crypto-map)# **set mtu** *length*	Sets the fragment size in tunnel mode.
Ruijie(config-crypto-map)# **set vrf** *vrf-name*	Specifies the VRF that are associated with tunnel.
Ruijie(config-crypto-map)#**username** *name* **password** *pass*	Configures the username and password used for extended authentication.
Ruijie(config-crypto-map)# **exit**	Exits crypto map configuration mode and returns to the global configuration mode.

- **Adding a dynamic crypto map set to the normal (static) crypto map set**

You can add one or more crypto map sets to a static crypto map set through reference of the crypto map entry to the dynamic map set. The crypto map entry that refers to the dynamic crypto map should be set as the entry with the lowest priority in the crypto map set.

To add a dynamic crypto map set to a static crypto map set, run the following command in global configuration mode:

Command	**Function**

Ruijie (config)# **crypto map** *map-name* *seq-num* **ipsec-isakmp dynamic** *dynamic-map-name*	Adds a dynamic crypto map set to a static crypto map set

Applying Crypto Map Entry to an Interface

To apply a crypto map set to an interface, run the following command in interface configuration mode:

Command	Function
Ruijie(config-if)# **crypto map** *map-name*	Applies a crypto map set to an interface.

A crypto map set should be configured for every interface that the IPSec traffic will pass. The device uses this crypto map set to judge all the traffic that passes this interface and apply a specific policy to filter the traffic.

Note

Only one crypto map set can be applied to an interface at one time, while the crypt map set can be applied to multiple interfaces at one time. When the IPSec traffic that passes this interface is processed, the IP address of this interface will be used as the address of the local device.

Creating Profile Crypto Map Entries

To create Profile crypto map entries for using IKE to establish SAs, run the following commands in global configuration mode in the beginning:

Command	Function
Ruijie(config)# **crypto map** *map-name seq-num* **ipsec-isakmp**	Specifies the Profile crypto map entry to be created or modified. When using this command, you will enter crypto map configuration mode.
Ruijie(config-crypto-map)# **match address** *access-list-id*	Specifies an access list for the crypto map list. This access list defines which communications are protected by the IPSec and which are not.
Ruijie(config-crypto-map)# **set peer** {*hostname* |*ip-address*} [*trustpoint1* [*trustpoint2*]]	Specifies remote IPSec peer. Communications protected by the IPSec are forwarding to this peer. Several peers can be configured.
Ruijie(config-crypto-map)# **set local** *ip-address*	Sets the IP address of the negotiation. Use the primary IP address of the interface if it is not otherwise configured.
Ruijie(config-crypto-map)# **set transform-set** *transform-set-name1* [*transform-set-name2…transform-set-name6*]	Specifies the transform set and lists the transform sets according to certain priority (set with higher priorities prevail).
Ruijie(config-crypto-map)# **set security-association lifetime seconds** *seconds*	
Or: Ruijie(config-crypto-map)# **set security-association lifetime kilobytes** *kilobytes*	

Command	Function
Ruijie(config-crypto-map)# **set exchange-mode** *main* | *aggressive*	Sets the mode to initiate negotiations by this static entry.
Ruijie(config-crypto-map)# **set pfs** *group1* | *group2*	Specifies the Diffie-Hellman group mark.
Ruijie(config-crypto-map)# **set mtu** *length*	Sets the length of pre-fragment in tunnel mode.

Repeat the preceding steps to create other necessary crypto map entries.

The following example shows how to configure to establish a SA using IKE:

Local peer (router A) configuration:

# Define a transform set named myset

```
crypto ipsec transform-set myset esp-des
```

# Define a map set named *profile-name* that establishes a SA using IKE.

```
crypto ipsec profile profile-name
set transform-set myset
```

Remote peer (router B) configuration:

# Define a transform set named myset.

```
crypto ipsec transform-set myset  esp-des
```

# Define a map set named *profile-name* that establishes a SA using IKE

```
crypto ipsec profile profile-name
set transform-set myset
```

Note

For the IPV6, IPSEC-IPV4, or IPSEC-IPV6 tunnels, the **match any** command must be configured in the map set *profile-name*. In addition, **Profile map** in this command can apply only to IPIP and IPv6 tunnels.

The dhcp over ipsec configuration takes effect only on the crypto map entry with the smallest value of *seq-num*.

Applying Profile Crypto Map Entries to a Tunnel Interface

To apply a Profile crypto map set to a tunnel interface, run the following command in interface configuration mode:

Command	Function
Ruijie`(config-if-Tunnel 1)#` **tunnel protection ipsec profile** *profile-name*	Applies a crypto map set to a tunnel interface.

A crypto map set must be configured for every interface IPSec traffic passes through. Then the device can use the crypto map set to decrypt all packets through these interfaces.

Note

Profile crypto map entries can apply only to a tunnel interface. An attempt to apply a Profile crypto map entry to a non-tunnel interface may fail. In addition, only GRE, IPIP, and IPv6 tunnels are supported. If the **match any** command is configured in an entry, the entry applies only to IPIP and IPv6 tunnels.

Configuring Extended Authentication

Extended authentication uses AAA authentication items to verify the identities of users. To configure this function, run the following command in configuration mode:

Command	Function
Ruijie `(config)#` **crypto map** *map-name* **client authentication list** *aaa-name*	Uses AAA authentication to verify identities of users.

Configuring Accounting Mode of Extensible Authentication

When the extensible authentication is implemented by using Authentication, Authorization and Accounting (AAA)/Remote Authentication Dial In User Service (RADIUS), some RADIUS servers require accounting packets for the keepalive purpose. In this case, the accounting mode of the extensible authentication needs to be configured. To do so, run the following command in configuration mode:

Command	Function
Ruijie `(config)#` **crypto map** *map-name* **client accounting list** *aaa-name*	Applies the AAA accounting list.

Note

After this command is configured, accounting packets are used only for the keepalive purpose, and accounting data such as traffic will not be updated.

Configuring IPSec Packet Filtering

This function determines whether decrypted original IPSec packets need to be filtered. To configure this function, run the following command in configuration mode:

Command	Function
Ruijie `(config)#` **crypto ipsec no-filter**	Decrypted packets are not filtered.

Monitoring and Maintaining IPSec

Some changes to the configuration only take effect when subsequent SAs are negotiated. If you want the new settings to take effect immediately, you must delete existing SAs, so that they will be established again using the new settings. The manually established SAs must be deleted and established again. Otherwise, changes will never take effect. If the device is processing the IPSec traffic, you may just want to clear the content that may be affected by the configuration change from the SA database (that is, only delete the SAs established by a given crypto map set). All the contents are only cleared from the SA database when the configuration is changed significantly, or the amount of the IPSec traffic that the device is processing is very small.

To delete and initiate the IPSec SA again, run the following commands in global configuration mode:

Command	**Function**
Ruijie# **clear crypto sa**	Clears the entire SA database. This will also delete all the active security threads.
Ruijie# **clear crypto sa peer** {*ip-address* | *peer-name*}	Clears the SAs with specific peer addresses.
Ruijie# **clear crypto sa map** *map-name*	Clears the SAs of a specific crypto map set.
Ruijie# **clear crypto sa spi** *destionation-address* {ah | esp} *spi*	Clears the SAs with the specified destination address, protocol, or SPI.

To view configuration information of IPSec, run the following command in normal user mode:

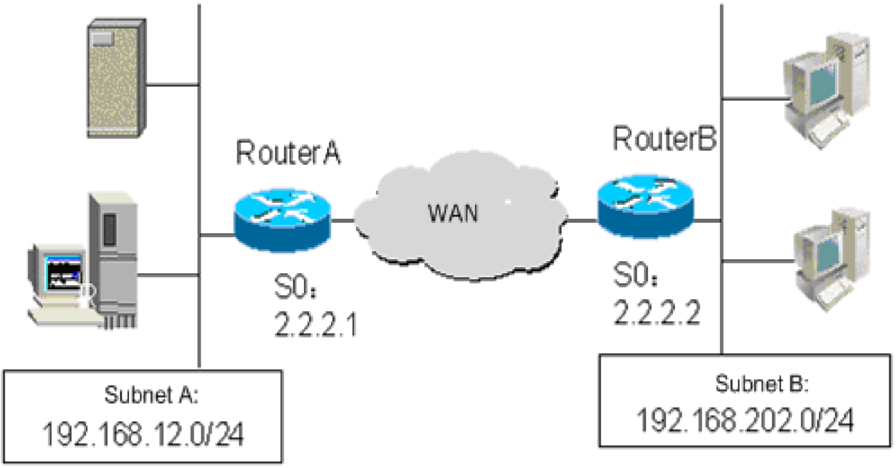
Command	**Function**
Ruijie# **show crypto ipsec transform-set**	Views configuration of a transform set.
Ruijie# **show crypto map** [*map-name*]	Views configuration of all or the specified crypto maps.
Ruijie# **show crypto ipsec sa**	Views information about the IPSec SA.
Ruijie# **show crypto dynamic-map** [*tag map-name*]	Views information about the dynamic crypto map.
Ruijie# **debug crypto ipsec**	Displays debug messages about the IPSec event.

IPSec Configuration Example

Configuration requirements

As shown in Figure 8, in order to protect the IP traffic from Subnet A (192.168.12.0/24) to Subnet B (192.168.202.0/24), use the Ethernet interface (192.168.12.1) of Router A and the Ethernet interface (192.168.202.1) of Router B as the security gateways at both ends, using the channel mode and the protection mode ESP-DES-SHA (the encryption and authentication services are available).

Figure 8 IPSec configuration example



Router configuration

In order to protect the IP traffic between hosts in two subnets, you can use the manually established SA or the IKE-established SA. Because Router A and Router B have similar configurations, the following only shows the configuration of the manually established SA for Router A. For the configuration of the IKE-established SA, refer to the typical case in the "IKE Configuration Guide" chapter .

Configuration of Router A:

# Use an access list to define the traffic to be protected

```
access-list 101 permit ip 192.168.12.0 0.0.0.255 192.168.202.0 0.0.0.255
```

# Define a transform set:

```
crypto ipsec transform-set myset esp-des esp-sha-hmac
```

# The crypto map associates the IPSec access list with the transform set, and specifies the destination of the protected traffic

```
crypto map mymap 10 ipsec-manual
    match address 101
    set transform-set myset
    set session-key inbound esp 301 cipher 0123456789abcdef authenticator
0000111122223333444455556666777788889999
set session-key outbound esp 300 cipher 0123456789abcdef authenticator
5555666677778888999900001111222233334444
    set peer 2.2.2.2
!
interface Fastethernet 0
  ip address 192.168.12.1 255.255.255.0
```

# Apply the crypto map to the interface

```
interface Serial 0
ip address 2.2.2.1
crypto map mymap
!
```

```
    ip route 0.0.0.0 0.0.0.0 2.2.2.2
```

IKE Configuration

IKE Working Process

IKE is a key management protocol standard that is used with the IPSec standard. IPSec is an IP security function that provides robust authentication and IP packet encryption. IPSec can be configured without using IKE. However, IKE enhances the IPSec function by providing additional functions and flexibility and making it easier to configure the IPSec standard. IKE is a hybrid protocol that implements the Oakley key exchange and the Skeme key exchange (ISAKMP, Oakley and Skeme are security protocols implemented by IKE) within the Internet Security Association and Key Management Protocol (ISAKMP) framework.

IPSec must be configured (depending on IPSec of IKE) and applied to the interface before IKE can work. When an outgoing packet that meets requirements is detected on the interface, IPSec will trigger IKE to negotiate with the peer IKE. They establish a secure channel between the peers to transmit various supported IPSec parameters. Finally, a consistent SA is established at both ends to enable IPSec at both ends to work. If there are data that meets requirements to be transmitted when the lifetime of the IPSec SA expires over time, IKEs at both ends will start to negotiate IPSec again, and so on.

IKE can be used to eliminate the need to manually specify all the IPSec parameters and keys in the crypto map tables for the two parties in communication. It allows you to specify the lifetime of the IPSec SA. IKE makes IPSec change the key regularly and thus strengthen the security. IKE enables IPSec to provide the anti-replay service.

IKE Configuration Task

IKE configuration tasks including:

Enabling or Disabling IKE: make sure IKE is working.

Ensuring Compatibility between Access List and IKE: if an access list (firewall) is configured on the device, you must make sure that the UDP packets of IKE are not prohibited.

Creating IKE Policies: specify parameters in each IKE policy.

Selecting Working Mode: There are two working modes-main mode (default) and aggressive mode. (note: aggressive mode is also called violent mode).

Configuring Local Identity: specifiy local indeidentity for IKE negotiation.

Setting Automatic Mode Recognition: specify whether the IKE negotiation responder automatically accept the negotiation in aggressive mode.

Configuring Digital Certificate: a digital certificate for IKE authentication.

Configuring Pre-shared Key: the pre-shared key is shared by the two peers participating in IKE negotiation.

Configuring DPD Detection: two mechanisms are used to implement DPD-- on-demand and periodic.

Configuring NAT Traversal Timeout: the UDC header is added to solve the NAT traversal problem. Use the keepalive packets to maintain the UDP linkage and to avoid the NAT connection timeout.

Exlude Ruijie vendor information: Ruijie vendor information is often delivered during IKE negotiation. If incompatibility is found in vendor informationIKE Session limit, exlude Ruijie vendor information

IKE SESSION LIMIT: set a limit on the number of IKE sessions. .

Ruijie Networks' IKE session mode: except for Digital signature authentication, switch all IKE negotiation within the network to Ruijie Networks' IKE session mode.

Extended Authentication Timeout: configure extended authentication timeout,

Configure domain authentication: confgure extended domain authentication to associate IPSEC tunnel with VRF.

Configure cisco's compatible extended authentication: configure this command on devices that negotiate with cisco's devices.

Exclude designated IP addresses from extended Digital signature authentication.

TRACK Correlation: in a scenario where there the primary and backup linkage exist at the same time, the IPSec is used to monitor the status of primary linkage. When the primary link is UP, the IPSec channel in the backup link will be removed automatically. Currently, we can use TRACK and DLDP to monitor the primary linkage.

Backup Link Detection: when the backup linkage detection does not work, the negotiation process will be intervened.

IKE Maintenance (optional): maintain IKE, make sure IKE is working, chech parameters.

Enabling or Disabling IKE

IKE is enabled by default. If you do not want to use IKE with IPSec together, you can disable it with a command. However, only the manual IPSec SA can work.

To disable or enable IKE, run the following command in global configuration mode:

Command	Function
Ruijie(config)# **no crypto isakmp enable**	Disables the IKE function.
Ruijie(config)# **crypto isakmp enable**	Enables the IKE function.

If IKE is disabled, subsequent configuration will not take effect.

Caution

Disabling IKE will result in:

- During IPSec communication, the encryption key never changes, and you need to modify your key regularly.
- The anti-replay service is unavailable.
- You must establish all the SAs manually.

Ensuring Compatibility between Access List and IKE

IKE is an application running on the basis of UDP that transmits packets in the UDP format through port 500. If an access list (firewall) is configured on the device, and UDP packets are prohibited, IKE negotiation will fail. Therefore, you must make sure that the UDP packets of IKE are not prohibited.

Creating IKE Policies

Both parties participating in IKE negotiation must have at least one set of consistent IKE policies. This is mandatory for successful IKE negotiation. You must create multiple policies with priorities on each peer, so as to make sure at least one policy matches the policy of the remote peer.

Define the following five parameters in each IKE policy:

Parameter	Keyword	Value Range	Default Value
Encryption algorithm	Des	56-bit DES-CBC	56-bit DES-CBC
	3des	168-bit DES-CBC	
	aes	128-bit AES-CBC	
HASH algorithm	Sha	SHA-1 (HMAC variant)	SHA-1 (HMAC variant)
	md5	MD5 (HMAC variant)	
Authentication method	pre-share	Pre-shared key	Digital signature authentication
	rsa-sig	Digital signature authentication	
	digital-email	Digital envelope authentication	
Diffie-Hellman group ID	1	768-bit Diffie-Hellman group	768-bit Diffie-Hellman group
	2	1024-bit Diffie-Hellman group	
	5	3072-bit Diffie-Hellman group	
	14	2048-bit Diffie-Hellman group	
	15	3072-bit Diffie-Hellman group	
	16	4096-bit Diffie-Hellman group	
	17	6144-bit Diffie-Hellman group	
	18	8192-bit Diffie-Hellman group	
Lifetime of IKE SA	Void	1 minute to 1 day (in seconds)	1 day (86400 seconds)

When IKE negotiation starts, IKE tries to find a consistent policy on the two peers. The initiator of negotiation sends all the policies to the remote responder. The responder searches, by priority, the policies that match with the local policies in the policies received from the remote peer.

If both peers participating in negotiation have the same encryption, HASH, authentication and Diffie-Hellman parameters, and the lifetime specified by the policy of the remote peer is smaller than or equal to the lifetime specified by the compared policy, then they match (if no lifetime is specified, the shorter lifetime specified by the policy of the remote peer is used). If no acceptable matching policy is found, IKE refuses to negotiate and IPSec is not established. If a matching policy is found, IKE negotiates and establishes the IPSec SA.

You can select a value of each parameter by making a tradeoff between security and performance:

- Encryption algorithms: At present, 56-bit DES-CBC, 168-bit 3DES-CBC, and 128-bit AES-CBC are supported.
- Hash algorithms: SHA-1 and MD5. MD5 has less digest and is often considered a little faster than SHA-1. Attacks on MD5 are proved to be successful in one way, but extremely difficult. However, the HMAC variant (MD5) used by IKE can block this attack.
- Authentication method: Currently RGOS supports pre-shared key method and digital certificate authentication. To authenticate with the pre-shared key method, you need to configure a correct pre-shared key. To authenticate using a digital certificate, you need to configure a correct certificate for both parties (refer to the section regarding certificate configuration).
- There are two options for the Diffie-Hellman group ID: 768-bit or 1024-bit Diffie-Hellman. 1024-bit Diffie-Hellman is more difficult to crack, but occupies more CPU resources.
- The lifetime of the IKE SA, different from that of the IPSec SA, refers to the validity period of IKE negotiation. It can be set to any value. The following is a general rule: The shorter the lifetime (to a critical point), the securer the IKE negotiation is. If a longer lifetime is used, however, negotiation of IPSec SA will be faster.

You can create multiple IKE policies, each of which corresponds to a different combination of parameters. A unique priority (1-10000, where 1 represents the highest priority) should be assigned to each created policy.

You can configure multiple policies on each peer. However, you must make sure that one of these policies has exactly the same encryption, HASH, authentication and Diffie-Hellman parameters as the remote peer (they can have different lifetimes). If no policy is configured, the device uses the default policy, which is set to have the lowest priority and includes the default value of each parameter.

To configure a policy, run the following commands in global configuration mode in the beginning:

Command	**Function**
Ruijie(config)# **crypto isakmp policy priority**	Identifies the policy to be created. Each policy is uniquely identified by the priority.
Ruijie(config-isakmp)# **encryption des | 3des | aes-128 | aes-192 | aes-256**	Specifies an encryption algorithm.
Ruijie(config-isakmp)# **hash {sha | md5}**	Specifies a HASH algorithm.
Ruijie(config-isakmp)# **authentication {pre-share | rsa-sig }**	Specifies an authentication method.
Ruijie(config-isakmp)# **group {1 | 2| 5}**	Specifies a Diffie-Hellman group ID.
Ruijie(config-isakmp)# **lifetime** *seconds*	Specifies the lifetime of IKE SA.
Ruijie(config-isakmp)# **exit**	Returns to global configuration mode.

If no value is specified for parameters, the default values are used.

To view the configured IKE policy, run the following command in privileged user mode:

Command	**Function**
Ruijie# **show crypto isakmp policy**	Shows all the existing IKE policies.

Note The configuration does not include the default policy and the default values of configured policies. To view these settings, use the **show crypto isakmp policy** command.

Selecting Working Mode

There are two working modes: main mode (default) and aggressive mode.

A working mode should be configured for the initiator. To do so, run the following commands at the crypto map entry where IKE is configured to establish a SA:

Command	**Function**
Ruijie(config)# **crypto map** *map-name seq-num* **ipsec-isakmp**	Specifies the crypto map entry to be created or modified. When using this command, you will enter crypto map configuration mode.
Ruijie(config-crypto-map)# **set exchange-mode {main | aggressive}**	Selects a working mode of IKE negotiation. The main mode is used by default.

By default, the responder negotiates in main mode. To use the aggressive mode, run the following command in global configuration mode.

Command	**Function**
Ruijie(config)**# crypto ipsec-isakmp mode-detect**	Negotiates in the mode used by the initiator.

To view the working mode of IKE, run the following command in privileged user mode:

Command	**Function**
Ruijie**# show crypto isakmp sa**	Browses all the current IKE SAs.

Configuring Local Identity

When selecting a working mode, you set the mode (main or aggressive mode) in which the initiator initiates the first negotiation message. If the main mode is selected, this configuration will not affect negotiation. If the aggressive mode is selected, this configuration specifies the identity type in the first negotiation message of the initiator, which directly affects negotiation in aggressive mode. Currently, you can set three forms: 1. Local address; 2. Domain name; 3. User name@domain name. You can set as necessary.

Command	**Function**

Ruijie(config)# **self-identity address** | **fqdn** | **user-fqdn** *identity* | dn	Specifies the form of the negotiation identity in aggressive mode. Address: The primary IP address of the local interface through which negotiation is initiated. Fqdn: Specifies the domain name form for the local identity. User-fqdn: Specifies the user name@domain name form for the local identity. Dn: DN value of the certificate

Setting Automatic Mode Recognition

The device, as the center, needs to accept dial-in in multiple modes (main mode and aggressive mode). The device needs to respond to the two different types of initiation and negotiate. Therefore, this command is mainly used in this working environment. Configuration of this command for the initiator has no influence.

Command	**Function**
Ruijie(config)**# crypto isakmp mode-detect**	Specifies that the responder negotiates by using automatic recognition.

Configuring Digital Certificate

By default, IKE authenticates by using a digital certificate. A digital certificate must be configured if this method is used. Refer to the "Digital Certificate Configuration" chapter.

Configuring Digital Envelope

The digital envelope authentication meets the need of IKE *IPSEC VPN Technology Specification.* The method is nearly the same with digital certificate. The different is when using digital envelope authentication, user should configure peer certificate first.

Here are key configurations, when configuring digital envelope:

Command	**Function**
Router(config)# **crypto pki certificate peer** *address*	Import peer certificate.
Ruijie(config)# **crypto isakmp policy** *Priority*	Identifies the policy to be created. Each policy is uniquely identified by the priority.
Ruijie(config-isakmp)# **authentication digital-email**	Specifies an authentication method.

Here are key configurations, when configuring new state code digital envelope authentication:

Command	**Function**
Router(config)# **crypto pki certificate peer** *address*	Import peer certificate.
Ruijie(config)# **crypto isakmp policy** *Priority*	Identifies the policy to be created. Each policy is uniquely identified by the priority.
Ruijie(config-isakmp)# **authentication digital-email asymmetric sm2**	Specifies new state code authentication mode.

Configuring Pre-shared Key

The pre-shared key is shared by the two peers participating in IKE negotiation. Therefore, each pre-shared key corresponds to a pair of IKE peers. On a given peer, you should specify a key that is the same as those of multiple pre-shared remote peers. For the purpose of security, you should configure different keys for different peer pairs.

To configure a pre-shared key, run the following commands in global configuration mode:

Command	Function
Router(config)# **ip host** *hostname address*	If hostname is used to identify the remote peer, specify the IP address corresponding to this hostname.
Ruijie(config)# **crypto isakmp key 0|7** *keystring* { **hostname** *peer-hostname* | **address** *peer-address* } [**no-xauth**]	Specifies a pre-shared key that is used with the remote IKE peer. Number 0 indicates the plain text is used. Number 7 indicates the cipher text is used.
Router(config)# **crypto isakmp key 0|7** *keystring* **address** *peer-address* [*mask*] [**no-xauth**]	Specifies a pre-shared key used for the IKE peer of a certain network segment. Both *peer-address* and *mask* are 0.0.0.0. 0.0.0.0 is the default pre-shared key.

You must configure the same pre-shared key on each pair of peers.

Note

1. Like Cisco devices, versions later than RGOS 8.31 use the digital signature authentication in the IKE policy by default. If the pre-shared key is needed, IKE policy configuration must be added. See the following example:

```
crypto isakmp policy 1
 authentication pre-share
!
```

Earlier versions of RGOS only support authentication using the pre-shared key.

2. If the hostname of a remote IKE negotiation peer has been registered on DNS, the second step mentioned above can be omitted.

3. On Cisco devices, if the peer uses the hostname to identify the pre-shared key, the initiator will initiate negotiation in aggressive mode.

4. After the extended authentication command **crypto map** *map-name* **client authentication list** *aaa-name* is configured, use the command **no-xauth** to disable the extended authentication for devices with designated address.

Configuring DPD Detection

Currently, two mechanisms are used to implement DPD: 1. on-demand. This mechanism sends packets when a tunnel is idle in a time longer than what is configured. This will trigger sending of a DPD message. 2. periodic. This mechanism actively sends a DPD message when the idle time of the tunnel exceeds the configured time. The maximum number of retransmission times is 5. (For DPD configuration in earlier versions, refer to the version 8.2 configuration guide)

To configure DPD detection, run the the following commands:

Command	Function
Router(config)# **crypto isakmp keepalive** *seconds*	seconds - Idle time of the tunnel The default retransmission interval is 5 seconds, and the on-demand mechanism is used
Router(config)# **crypto isakmp keepalive** *seconds retries*	seconds - Idle time of the tunnel retries – Retransmission interval The on-demand mechanism is used by default.
Router(config)# **crypto isakmp keepalive** *seconds retries* **on-demand**	seconds - Idle time of the tunnel retries - Retransmission interval on-demand - The on-demand mechanism
Router(config)# **crypto isakmp keepalive** *seconds* **periodic**	seconds - Idle time of the tunnel periodic - periodic mechanism The default Retransmission interval is 5 seconds.
Router(config)# **crypto isakmp keepalive** *seconds retries* **periodic**	seconds - Idle time of the tunnel retries - Retransmission interval periodic - periodic mechanism

Configuring NAT Traversal Timeout

As the RFC3947 and IPSEC NAT-t are supported, the UDC header is added to solve the NAT traversal problem. To avoid the NAT connection timeout, the keepalive mode shall be used to send packets. The default time is 5 minutes.

Command	Function
Router(config)# **crypto isakmp nat keepalive** *seconds*	Seconds: the interval of sending the packets in keepalive mode. The default interval is 5 minutes.

Excluding Ruijie Vendor Information

Ruijie vendor information is often delivered during IKE negotiation. If incompatibility is found in vendor information, run the following command:

Command	Function
Router(config)# **crypto isakmp vendorid disable**	Excludes the vendor id information.

Configuring IKE Session Limit

To set a limit on the number of IKE sessions, run the following command:

Command	Function
Router(config)# **crypto isakmp session limit number**	Sets a limit on the number of IKE sessions.

Configuring Ruijie IKE Negotiation Mode

To switch all IKE negotiation (except for Digital signature authentication) within the network to Ruijie Networks' IKE session mode, run the following command:

Command	Function
Router(config)# **crypto isakmp rg-sm1**	Switches all IKE negotiation (except for Digital signature authentication) within the network to Ruijie Networks' IKE session mode

Configuring Extended Authentication Timeout

To configure extended authentication timeout, run the following command:

Command	Function
Router(config)# **crypto isakmp xauth timeout** *seconds*	Configures the timeout time of extended authentication, with the value ranging from 5 to 90 seconds.

Configuring AAA Server Response Timeout

To configure AAA server response timeout, run the following command:

Command	Function
Router(config)# **crypto isakmp xauth timeout** *seconds*	Configures the timeout time of waiting AAA server response, with the value ranging from 5 to 10000 seconds.

Configuring Client Policy Delivery

When both Key ID authentication and extended authentication are used on a client, run the following commands in configuration mode to configure client policy delivery:

Command	Function
Ruijie(config)# **crypto isakmp client configuration group** name	Creates or modifies a client configuration delivery entry. When using this command, you will enter client policy delivery configuration mode.
Router(config-isakmp-group)# **key 0|7** *keystring*	Configures the shared key used for Key ID authentication. This configuration takes effect only in aggressive mode.
Router(config-isakmp-group)# **dns** pri-dns sec-dns	Configures the DNS from which a policy is delivered to a client.
Router(config-isakmp-group)# **netmask** mask	Configures the subnet mask from which a policy is delivered to a client.
Router(config-isakmp-group)# **pool** pool-name	Configures the address pool from which an IP address is selected for delivering a policy to a client.

Router(config-isakmp-group)# **network center** *net-addr/prefix*	(optional) open for the networks under the client's server. Only tht packeks forwarding to these networks can pass over the IPSec tunnel. Currently, the maximum unmber of these nuetworks is 5.

Configuring domain authentication

To enable domain authentication:

Command	**Function**
Router(config)# **crypto isakmp authorize** [**split**]	enables domain authentication:

To configure domain-delimiter:

Command	**Function**
Router(config)# **crypto isakmp domain-delimiter** *keyword* [**prefix| suffix**]	Specifies domain-delimiter；by default, the **suffix** is domain.

To specify domain-name ,

Command	**Function**
Router(config-isakmp-group)# **domain** *domain-name* [**vrf**] *vrf-name*	Specifies the domain-name and associates domain-name with vrf-name

Configuring cisco's compatible extended authentication:

To adopt cisco's compatible extended authentication for IKE negotiation. Run the following commands in configuration mode:

Command	**Function**
Router(config)# **crypto isakmp xauth cisco_comp**	Adopts Cisco's compatible extended authentication.

Note

After configured the command **crypto map** *map-name* **client authentication list** *aaa-name* on a crypto map, all clients need to be authenticated. However, some client with designated IP address do not need extended authentication. To exclude these designated IP addresses from extended Digital Signature Authentication, run the commend mentioned in the above table.

Configuring IP address pool

Use the following commands to issue an IP address for a XAUTH client:

Command	**Function**
Router(config)# **crypto isakmp ippool** *pool-name*	Creates an address pool
Ruijie(config-isakmp-ippool)#**address** *low-ip high-ip*	Configures the range of the IP address pool.

TRACK Correlation

In a scenario where there is a backup linkage or multiple linkage, the IPSec is used to monitor the status of primary linkage. When the primary linkage is up, the IPSec channel in the backup linkage will be removed so as to clear the reverse routing. And then the normal data forwarding is guranteed.Currently, TRACK and DLDP are used to monitor primary linkage. Refer to the corresponding files for the configuration of TRACK and DLDP.

Command	Function
Ruijie(config)#**crypto isakmp link-redundency backup** *backup_interface* **track** *track_id*	Monitors *track id* via the TRACK protocol. The IPSEC channel on *backup_interface* will be removed after *track id* is up.
Ruijie(config)#**crypto isakmp link-redundency backup** *backup_interface* **dldp** *master_interface*	Monitors *master_interface* via the DLDP protocol. The IPSEC channel on *backup_interface* will be removed after the *master_interface* is up.
Ruijie(config)#**crypto isakmp link-redundency backup** *backup_interface* **intf-down** *master_interface*	Monitors the DOWN event in *master_interface*. The IPSEC channel on *backup_interface* will be removed after it occurs.

Backup Linkage Detection

When the default backup linkage detection does not work, the negotiation process will be intervened

Command	Function
Ruijie(config)#**crypto isakmp link-redundency detect ike**	Only negotiate the first stage of IKE and ensure its basic function to be normal.

⚠ Caution

the command will only be configured when the default detection does not work. The default detection will detect the backup linkage in a better way.

IKE Maintenance

Clear an IKE connection

To clear an IKE connection, run the following commands in privileged user mode:

Command	Function
Ruijie# **show crypto isakmp sa**	Shows the existing IKE connections, and note down the connection ID of the connection you want to clear.
Ruijie# **clear crypto isakmp** [*connection-id*]	Clears an IKE connection. When connection-id is not used, all IKE connections are cleared.
Directly execute: Ruijie# **clear crypto isakmp**	Clears all the local IKE connections.

IKE diagnosis

To obtain the IKE diagnostics information, run the following commands in privileged user mode:

Command	Function
Ruijie# **show crypto isakmp policy**	Shows all the IKE policy parameters.
Ruijie# **show crypto isakmp sa**	Shows all the current IKE SAs.
	Shows IKE address pool
Ruijie# **debug crypto isakmp**	Shows debug messages about the IKE event.
	Shows debug message about the IKE address pool event.

IKE Configuration Example

The following shows an IKE configuration example:

```
crypto isakmp enable
crypto isakmp policy 4
    group 1
    encryption 3des
crypto isakmp policy 5
  authen pre-share
  group 2
  lifetime 1000
crypto isakmp policy 6
  authen rsa-sig
  group 1
  lifetime 1000
For details, run the following command in privileged user mode:
Ruijie # show crypto isakmp policy
Protection suite of priority 4
encryption algorithm:  3DES - Data Encryption Standard (56 bit keys).
  hash algorithm:        Secure Hash Standard
  authentication method: Rsa-Sig
  Diffie-Hellman group:  #1 (1024 bit)
  lifetime:             1000 seconds
Protection suite of priority 5
  encryption algorithm:  DES - Data Encryption Standard (56 bit keys).
  hash algorithm:        Secure Hash Standard
  authentication method: Pre-shared key
  Diffie-Hellman group:  #2 (1024 bit)
  lifetime:             1000 seconds
Protection suite of priority 6
  encryption algorithm:  DES - Data Encryption Standard (56 bit keys).
  hash algorithm:        Secure Hash Standard
  authentication method: Rsa-Sig
  Diffie-Hellman group:  #1 (768 bit)
  lifetime:             1000 seconds
Default protection suite
```

```
  encryption algorithm:  DES - Data Encryption Standard (56 bit keys).
  hash algorithm:        Secure Hash Standard
  authentication method: Rsa-Sig
  Diffie-Hellman group:  #1 (768 bit)
  lifetime:              86400 seconds
```

Typical Application Cases

Statically Configuring Tunnels

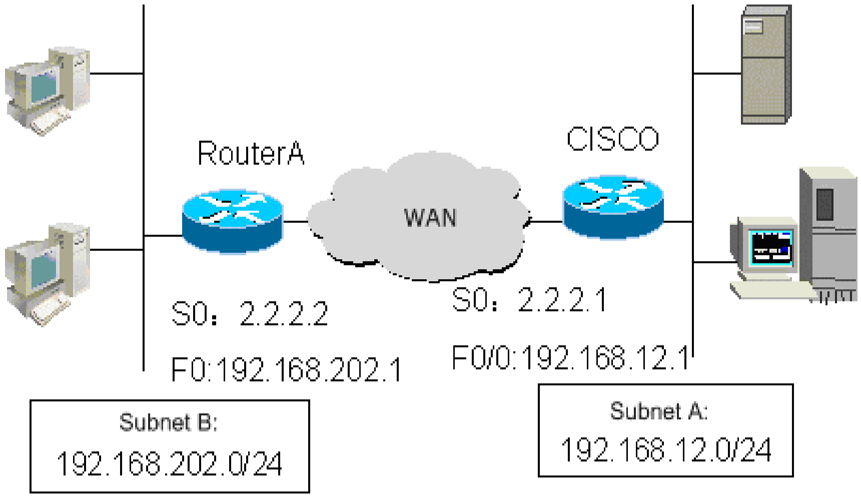
Analysis

In this case, the IP traffic between two subnets is protected by using a Cisco device connected to Subnet A as the gateway at one side, and using a Ruijie device connected to Subnet B as the gateway at the other side. The following requirements should be met:

- The 3DES algorithm is used in phase 1.
- The tunnel mode is used.
- The protection method is ESP-DES-MD5 (the encryption and authentication services are available).

In the following application, a Cisco device is used as the center, and Ruijie devices are used as remote branches. See Figure 9 :

Figure 9 Typical IPSec application



Router configuration

This section describes how to manually establish a SA and establish a SA using IKE for Ruijie router, namely Router A. Meanwhile, it also provides configurations of Cisco devices for your reference in actual work.

8) Configuration for establishing a SA using IKE

Configuration of Router A:

```
!
hostname "RouterA"
```

# Enable IKE

```
crypto isakmp enable
crypto isakmp policy 1
authentication pre-share
encryption 3des
!
```

# Configure a pre-shared key and a transform set

```
crypto isakmp key 0 preword address 2.2.2.1
crypto ipsec transform-set myset  esp-des esp-md5-hmac
```

# Define a crypto map set

```
crypto map mymap 5 ipsec-isakmp
 set peer 2.2.2.1
 set transform-set myset
 match address 101
!
interface FastEthernet0
 ip address 192.168.202.1 255.255.255.0
```

# Apply the crypto map to the interface

```
interface Serial0
 ip address 2.2.2.2 255.255.255.0
 encapsulation ppp
 crypto map mymap
!
ip route 0.0.0.0 0.0.0.0 Serial0
```

# Define an encryption access list to protect the IP traffic between the subnet 192.168.202.0/24 and the subnet 192.168.12.0/24

```
access-list 101 permit ip 192.168.202.0 0.0.0.255 192.168.12.0 0.0.0.255
!
end
```

Configuration of Cisco device:

```
!
hostname Cisco
```

# Define an IKE policy, using the pre-shared key for authentication, and using the default values for other parameters

```
crypto isakmp policy 1
 authentication pre-share
encryption 3des
```

# Configure a pre-shared key

```
crypto isakmp key 0 preword address 2.2.2.2
```

# Define a transform set

```
crypto ipsec transform-set myset esp-des esp-md5-hmac
```

# Define a crypto map

```
crypto map mymap 5 ipsec-isakmp
 set peer 2.2.2.2
 set transform-set myset
 match address 101
!
interface FastEthernet0/0
 ip address 192.168.12.1 255.255.255.0
```

# Apply the crypto map to the interface

```
interface Serial0
 ip address 2.2.2.1 255.255.255.0
 encapsulation ppp
 crypto map mymap
!
ip route 192.168.202.0 255.255.255.0 Serial0
```

# Define an encryption access list to protect the IP traffic between the subnet 192.168.12.0/24 and the subnet 192.168.202.0/24

```
access-list 101 permit ip 192.168.12.0 0.0.0.255 192.168.202.0 0.0.0.255
!
end
```

Configuration for establishing a SA manually

Configuration of Router A:

```
!
hostname "RouterA"
```

# Define a transform set

```
crypto ipsec transform-set myset  esp-des esp-md5-hmac
```

# Define a crypto map set

```
crypto map mymap 5 ipsec-manual
 set peer 2.2.2.1
 set session-key inbound esp 300 cipher abcdef1234567890
 authenticator abcdef1234567890abcdef1234567890  //This is the same configuration statement
as the previous line
 set session-key outbound esp 301 cipher abcdef1234567890
 authenticator abcdef1234567890abcdef1234567890  //This is the same configuration statement
as the previous line
```

```
 set transform-set myset
 match address 101
!
interface FastEthernet0
 ip address 192.168.202.1 255.255.255.0
```

# Apply the crypto map to the interface

```
interface Serial0
 ip address 2.2.2.2 255.255.255.0
 encapsulation ppp
 crypto map mymap
!
ip route 0.0.0.0 0.0.0.0 Serial0
```

# Define an encryption access list to protect the IP traffic between the subnet 192.168.202.0/24 and the subnet 192.168.12.0/24

```
access-list 101 permit ip 192.168.202.0 0.0.0.255 192.168.12.0 0.0.0.255
!
end
```

Configuration of Cisco device:

```
!
hostname Cisco
```

# Define a transform set

```
crypto ipsec transform-set myset esp-des esp-md5-hmac
```

# Define a crypto map

```
crypto map mymap 5 ipsec-manual
 set peer 2.2.2.2
 set session-key inbound esp 301 cipher abcdef1234567890
 authenticator abcdef1234567890abcdef1234567890  //This is the same configuration statement
as the previous line
 set session-key outbound esp 300 cipher abcdef1234567890
 authenticator abcdef1234567890abcdef1234567890  //This is the same configuration statement
as the previous line
 set transform-set myset
 match address 101
!
interface FastEthernet0/0
 ip address 192.168.12.1 255.255.255.0
# Apply the crypto map to the interface
interface Serial0
 ip address 2.2.2.1 255.255.255.0
 encapsulation ppp
 crypto map mymap
!
ip route 192.168.202.0 255.255.255.0 Serial0
```

# Define an encryption access list to protect the IP traffic between the subnet 192.168.12.0/24 and the subnet 192.168.202.0/24

```
access-list 101 permit ip 192.168.12.0 0.0.0.255 192.168.202.0 0.0.0.255
!
end
```

Monitoring and debugging

9) Monitoring and debugging the IKE-based SA

On any host in Subnet B, send a packet to Subnet A. IKE negotiation is triggered and finally an IPSec SA is established successfully.

# Turn on the debugging switches of IKE and IPSec:

```
RouterA# debug  crypto ipsec
IPSEC debugging is on
RouterA# debug  crypto isakmp
ISAKMP debugging is on
```

# You can see the following debugging information during negotiation:

```
Get acquire: 192.168.202.0/0.0.0.255 -> 192.168.12.0/0.0.0.255 , prot 0, port 0/0
Acqurire negotiate with 2.2.2.1
(36) Beginning Quick Mode exchange, M-ID of 4445127
(36) sending packet to 2.2.2.1 (I) QM_SI1_WR1
ipsec_output:423, get item acclist 101
ipsec_output:429, match  3
(36) received packet from 2.2.2.1 (I) QM_SI1_WR1
payload format: <Hdr>,<hash> <sa> <nonce> <id>
(36) processing SA payload. message ID = 4445127
(36) Creating IPSec SAs.
    inbound SA has spi 4445127
    protocol esp, DES_CBC
    auth MD5
    outbound SA has spi 275385850
    protocol esp, DES_CBC
    auth MD5
    lifetime of 3600 seconds, soft 3570 seconds
    lifetime of 4608000 kilobytes, soft 256 kilobytes
ipsec_output:423, get item acclist 101
ipsec_output:429, match  3
(36) sending packet to 2.2.2.1 (I) QM_IDLE
(36) Phase_2 negotiate complete!
```

# In order to view and confirm whether the SAs of IKE and IPSec have been established, use the following command:

```
RouterA# show crypto isakmp sa
destination    source         state                 conn-id
lifetime(second)
 2.2.2.1       2.2.2.2         QM_IDLE               36
5013
```

# The preceding information shows that an IKE SA has been established successfully

```
RouterA# show crypto ipsec sa
Interface: Serial0
Crypto map tag:mymap, local addr  2.2.2.2  //The current crypto map set is named mymap that
uses the local address 2.2.2.2
    media mtu 1500
    local  ident (addr/mask/prot/port): (192.168.202.0/0.0.0.255/0/0))
    remote  ident (addr/mask/prot/port): (192.168.12.0/0.0.0.255/0/0))
PERMIT                 //Protect the traffic between 192.168.202.0/24 and 192.168.12.0/24
current_peer: 2.2.2.1      //The address of the remote peer is 2.2.2.1
    #pkts encaps: 3, #pkts encrypt: 3, #pkts digest 3
    #pkts decaps: 3, #pkts decrypt: 3, #pkts verify 3
    #send errors 0, #recv errors 0
//Statistical data in turn: number of encapsulation packets, number of encryption packets,
number of digest packets, number of de-capsulation packets, number of decryption packets, number
of verification packets, send errors, and receive errors.
    inbound esp sas:                  //Security association for the inbound packet
processing, with the protocol ESP
            spi:0x43D3C7 (4445127)          //The SPI value is 4445127
        transform: esp-des esp-md5-hmac    //The transform set is esp-des-md5
             in use settings={Tunnel,}      //Tunnel mode
sa timing: remaining key lifetime (k/sec): (4607999/3578)
//There are 4607999 kbytes/3578 seconds left before expiry of the SA
             IV size: 8 bytes               //The IV vector length is 8
             Replay detection support:Y     //Anti-replay processing

    outbound esp sas:                      //Security association for the outbound packet
processing, with the protocol ESP
            spi:0x106A0DFA (275385850)      //The SPI value is 275385850
        transform: esp-des esp-md5-hmac    //The transform set is esp-des-md5
             in use settings={Tunnel,}      //Tunnel mode
sa timing: remaining key lifetime (k/sec): (4607999/3577)
//There are 4607999 kbytes/3577 seconds left before expiry of the SA
             IV size: 8 bytes               //The IV vector length is 8
             Replay detection support:Y     //Anti-replay processing
```

The preceding statistical data shows that IPSec has been established and some packets are protected.

Monitoring and debugging the manually established SA

Because the manually established SA exists from the beginning, you needn't negotiate it and cannot view its debugging information. You can only view some statistical data:

```
RouterA# show crypto ipsec sa
Interface: Serial0
Crypto map tag:mymap, local addr  2.2.2.2  //The current crypto map set is named mymap that uses the local address 2.2.2.2
    media mtu 1500
    local  ident (addr/mask/prot/port): (192.168.202.0/0.0.0.255/0/0))
    remote  ident (addr/mask/prot/port): (192.168.12.0/0.0.0.255/0/0))
PERMIT                               //Protect the traffic between 192.168.202.0/24 and 192.168.12.0/24
current_peer: 2.2.2.1                    //The address of the remote peer is 2.2.2.1
    #pkts encaps: 8, #pkts encrypt: 8, #pkts digest 8
    #pkts decaps: 8, #pkts decrypt: 8, #pkts verify 8
    #send errors 0, #recv errors 0
//Statistical data in turn: number of encapsulation packets, number of encrypted packets, number of digest packets, number of decapsulated packets, number of decryption packets, number of verification packets, send errors, and receive errors.
inbound esp sas:                     //Security association for the inbound packet processing, with the protocol ESP
spi: 0x12C (300)               //The SPI value is 300
transform: esp-des esp-md5-hmac    //The transform set is esp-des-md5
in use settings={Tunnel,}      //Tunnel mode
no sa timing                             //Expired
IV size: 8 bytes               //The IV vector length is 8
Replay detection support:N     //No anti-replay processing

outbound esp sas:                //Security association for the outbound packet processing, with the protocol ESP
spi: 0x12D (301)               //The SPI value is 301
transform: esp-des esp-md5-hmac    //The transform set is esp-des-md5
in use settings={Tunnel,}      //Tunnel mode
no sa timing                             //Expired
IV size: 8 bytes           //The IV vector length is 8
Replay detection support:N      //No anti-replay processing
```

The preceding statistical data shows that IPSec has been established and some packets are protected.

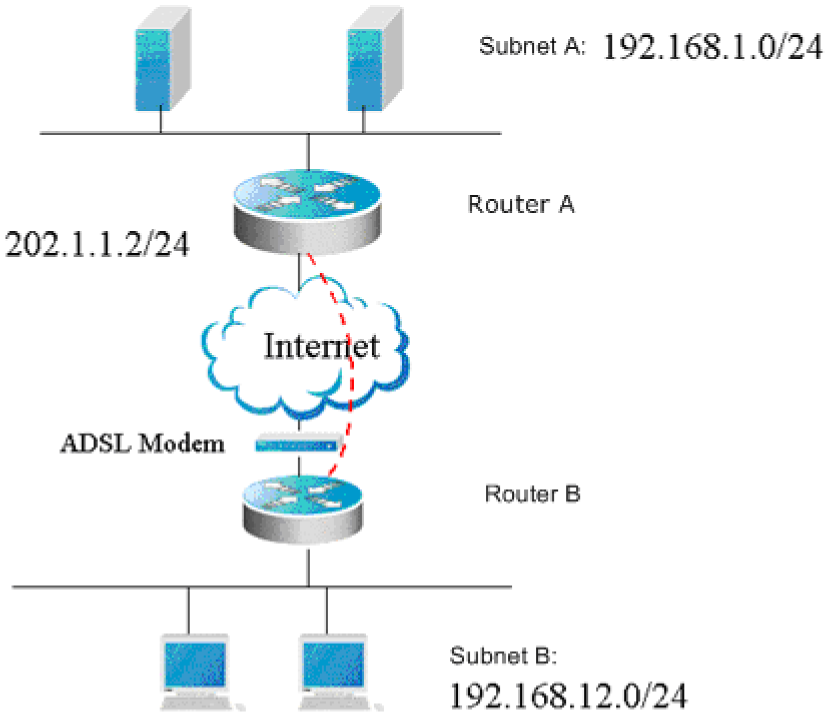
Dynamically Configuring Tunnels

Case analysis

In this case, the IP traffic between two subnets is protected by using Ruijie Router A connected to Subnet A as the gateway at one side, and using Ruijie Router B connected to Subnet B as the branch gateway at the other side, as shown in Figure-4. The following requirements should be met:

- The tunnel mode is used.
- The protection method is ESP-DES-MD5 (the encryption and authentication services are available).
- The IP address of the WAN interface of Router A is fixed: 202.1.1.2/24. The router is connected to the Internet through a dedicated line.
- Router B is connected to the Internet through ADSL using the PPPOE protocol. Its IP address is allocated by the ISP dynamically.
- Use IKE to establish the SA.
- Use the pre-shared key.

Figure 10 IPSec typical case



Router configuration

This section provides configuration for establishing a SA between Ruijie routers, namely Router B and Router A.

Configuration of Router B:

```
!
hostname "RouterB"
```

# Enable IKE

```
crypto isakmp enable
```

# Configure a pre-shared key and a transform set

```
crypto isakmp policy 1
 authentication pre-share
crypto isakmp key 0 preword address 202.1.1.2
crypto ipsec transform-set myset  esp-des esp-md5-hmac
```

# Define a crypto map set

```
crypto map mymap 5 ipsec-isakmp
 set peer 202.1.1.2
 set transform-set myset
 match address 101
!
interface FastEthernet0
 ip address 192.168.12.1 255.255.255.0
interface FastEthernet1
 no ip addrsss
 pppoe enable
 pppoe-client 1 dial-pool-number 1 dial-on-demand
```

# Apply the crypto map to the interface

```
interface Dialer0
  mtu 1488
ip address negotiate
  encapsulation ppp
  ppp pap sent-username xxx password xxx
  crypto map mymap
dialer idle-timeout 2400
  dialer pool 1
  dialer-group 1
!
dialer-list protocol ip permit
ip route 0.0.0.0 0.0.0.0 Dialer0 permanent
```

# Define an encryption access list to protect the IP traffic between the subnet 192.168.12.0/24 and the subnet 192.168.1.0/24

```
access-list 101 permit ip 192.168.12.0 0.0.0.255 192.168.1.0 0.0.0.255
!
end
```

Configuration of Router A:

```
!
hostname “RouterA”
```

# Define an IKE policy, using the pre-shared key for authentication, and using the default values for other parameters

```
crypto isakmp policy 1
 authentication pre-share
```

# Configure the default pre-shared key. Because the IP address of the remote end is dynamic, you couldn't know in advance it is necessary to configure the default pre-shared key

```
crypto isakmp key 0 preword address 0.0.0.0 0.0.0.0
```

# Define a transform set

```
crypto ipsec transform-set myset esp-des esp-md5-hmac
```

# Define a dynamic crypto map

```
crypto dynamic-map dymymap 5
 set transform-set myset
 match address 101
!
```

# Add a dynamic crypto map set to a static crypto map set

```
crypto map mymap 10 ipsec-isakmp dynamic  dymymap
!
interface FastEthernet0/0
 ip address 192.168.1.1 255.255.255.0
```

# Apply the crypto map to the interface

```
interface Serial0
 ip address 202.1.1.2 255.255.255.0
 encapsulation ppp
 crypto map mymap
!
ip route 0.0.0.0 0.0.0.0  Serial0
```

# Define an encryption access list to protect the IP traffic between the subnet 192.168.1.0/24 and the subnet 192.168.12.0/24

```
access-list 101 permit ip 192.168.1.0 0.0.0.255 192.168.12.0 0.0.0.255
!
end
```

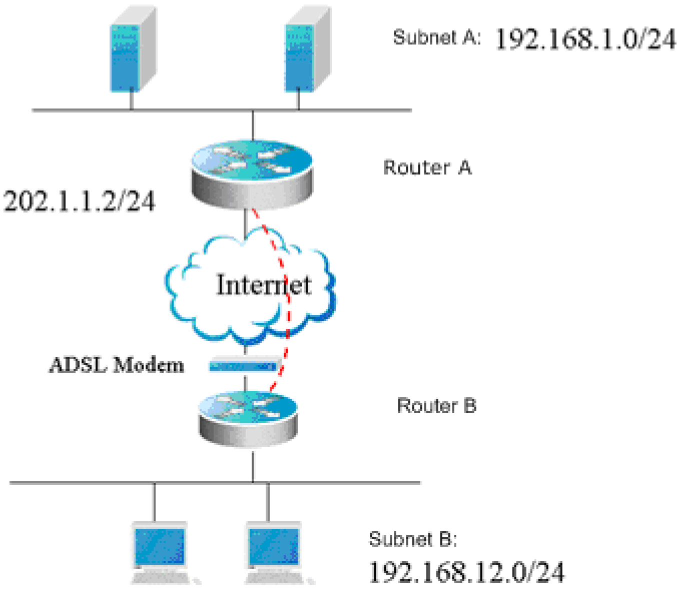
Initiating Negotiation with the Domain Name

Case analysis

In this case, the IP traffic between two subnets is protected by using Ruijie Router A connected to Subnet A as the gateway at one side, and using Ruijie Router B connected to Subnet B as the branch gateway at the other side. The following requirements should be met:

- The tunnel mode is used.
- The protection method is ESP-DES-MD5 (the encryption and authentication services are available).
- The IP address of the WAN interface of Router A is always 202.1.1.2/24. The router is connected to the Internet through a dedicated line.
- Router B is connected to the Internet through ADSL using the PPPOE protocol. Its IP address is allocated by ISP dynamically.
- Use the pre-shared key, and specify the pre-shared key for the central router using the host name.
- Use IKE to establish the SA.

Figure 11



Router configuration

This section describes how to establish a SA between Ruijie routers, namely Router B and Router A.

Configuration of Router B:

```
!
hostname "RouterB"
```

# Enable IKE

```
crypto isakmp enable
```

# Configure the local identity

```
self-identity fqdn www.google.com
```

# Configure a pre-shared key and a transform set

```
crypto isakmp key 0 preword address 202.1.1.2
crypto ipsec transform-set myset  esp-des esp-md5-hmac
```

# Define a crypto map set

```
crypto map mymap 5 ipsec-isakmp
 set peer 202.1.1.2
 set exchange-mode aggressive
 set transform-set myset
 match address 101
!
interface FastEthernet0
 ip address 192.168.12.1 255.255.255.0
interface FastEthernet1
 no ip addrsss
 pppoe enable
 pppoe-client 1 dial-pool-number 1 dial-on-demand
```

# Apply the crypto map to the interface

```
interface Dialer0
  mtu 1488
ip address negotiate
  encapsulation ppp
  ppp pap sent-username xxx password xxx
  crypto map mymap
dialer idle-timeout 2400
  dialer pool 1
  dialer-group 1
!
dialer-list protocol ip permit
ip route 0.0.0.0 0.0.0.0 Dialer0 permanent
```

# Define an encryption access list to protect the IP traffic between the subnet 192.168.12.0/24 and the subnet 192.168.1.0/24

```
access-list 101 permit ip 192.168.12.0 0.0.0.255 192.168.1.0 0.0.0.255
!
end
```

Configuration of Router A:

```
!
hostname “RouterA”
```

# Define an IKE policy, using the pre-shared key for authentication, and using the default values for other parameters

```
crypto isakmp policy 1
 authentication pre-share
```

# Configure a default pre-shared key. Because the IP address of the remote end is dynamic, the pre-shared key is found by specifying the hostname

```
crypto isakmp key 0 preword hostname www.google.com
```

# Configure automatic recognition for the center

```
crypto isakmp mode-detect
```

# Define a transform set

```
crypto ipsec transform-set myset esp-des esp-md5-hmac
```

# Define a dynamic crypto map

```
crypto dynamic-map dymymap 5
 set transform-set myset
 match address 101
!
```

# Add a dynamic crypto map set to a static crypto map set

```
crypto map mymap 10 ipsec-isakmp dynamic  dymymap
!
interface FastEthernet0/0
 ip address 192.168.1.1 255.255.255.0
```

# Apply the crypto map to the interface

```
interface Serial0
 ip address 202.1.1.2 255.255.255.0
 encapsulation ppp
 crypto map mymap
!
ip route 0.0.0.0 0.0.0.0  Serial0
```

# Define an encryption access list to protect the IP traffic between the subnet 192.168.1.0/24 and the subnet 192.168.12.0/24

```
access-list 101 permit ip 192.168.1.0 0.0.0.255 192.168.12.0 0.0.0.255
!
End
```

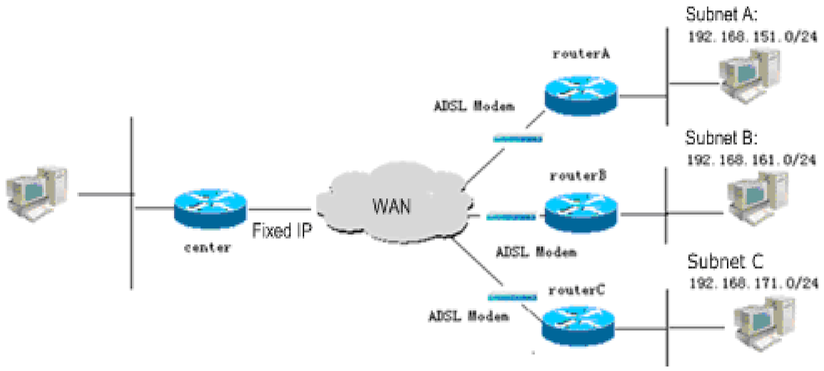
Dynamically Configuring to Use Certificate Negotiation

Case analysis

Configure a dynamic crypto map set for the center. The IKE negotiation policy uses the digital signature for authentication. The branch is connected to the center through L2TP. L2TP will trigger IKE negotiation. The IPSec tunnel must be established successfully before the L2TP tunnel can be established. L2TP is used to run OSPF, so that both the center and the branch can learn their own subnet routes. The following requirements should be met:

- The tunnel mode is used.
- The protection method is ESP-DES (the encryption service is available).
- The IP address of the WAN interface of the center is always 63.23.12.212/29. The center is connected to the Internet through a dedicated line.
- The branch router is connected to the Internet through ADSL using the PPPOE protocol. Its IP address is allocated by the ISP dynamically.
- The central router is configured with IKE negotiation policies and uses the certificate for authentication.

Figure 12



Configuration of Center:

```
hostname center
!
!
route-map static-ospf permit 10
 match ip address 1
!
access-list 1 permit 0.0.0.0 255.255.255.0
!
vpdn enable
!
vpdn-group 1
! Default L2TP VPDN group
 accept-dialin
  protocol l2tp
  virtual-template 1
 l2tp tunnel force_ipsec
source-ip 63.23.12.212
!
username RGOS password 0 RGOS
!
```

# Configure the certificate and root certificate for the local router

```
crypto pki certificate chain
 certificate ca 56AE073C10A17E8C45AE8D3F15523357
 3082032E 308202D8 A0030201 02021056 AE073C10 A17E8C45 AE8D3F15 52335730
 0D06092A 864886F7 0D010105 05003081 AC312130 1F06092A 864886F7 0D010901
 16126469 6E676A73 40737461 722D6E65 742E636E 310B3009 06035504 06130243
 4E310F30 0D060355 04081306 46754A69 616E310F 300D0603 55040713 0646755A
 686F7531 20301E06 0355040A 13175265 6769616E 74204E65 74776F72 6B20436F
 2E204C74 64311D30 1B060355 040B1314 52657365 61726368 20417061 72746D65
 6E742035 31173015 06035504 03130E43 41207465 73742073 65727665 72301E17
 0D303530 32323530 38343630 325A170D 30373033 30313032 33363233 5A3081AC
 3121301F 06092A86 4886F70D 01090116 1264696E 676A7340 73746172 2D6E6574
 2E636E31 0B300906 03550406 1302434E 310F300D 06035504 08130646 754A6961
```

```
6E310F30 0D060355 04071306 46755A68 6F753120 301E0603 55040A13 17526567
69616E74 204E6574 776F726B 20436F2E 204C7464 311D301B 06035504 0B131452
65736561 72636820 41706172 746D656E 74203531 17301506 03550403 130E4341
20746573 74207365 72766572 305C300D 06092A86 4886F70D 01010105 00034B00
30480241 00D91F1A C60EF951 924CDC96 4D4443FA DABE53F2 DDF513B6 34B5A6A1
9FFD57A1 6C6F7AF4 A113C159 3D0C4C3E 8E62DE76 D8A24CF2 2CF8DA82 AA17D3E8
CC80C295 8F020301 0001A381 D33081D0 300B0603 551D0F04 04030201 C6300F06
03551D13 0101FF04 05300301 01FF301D 0603551D 0E041604 14724384 1C2B0345
2D8258F5 844377F5 21AA4B2C 21307F06 03551D1F 04783076 3038A036 A0348632
68747470 3A2F2F7A 6A2D726F 75746572 2F436572 74456E72 6F6C6C2F 43412532
30746573 74253230 73657276 65722E63 726C303A A038A036 86346669 6C653A2F
2F5C5C7A 6A2D726F 75746572 5C436572 74456E72 6F6C6C5C 43412532 30746573
74253230 73657276 65722E63 726C3010 06092B06 01040182 37150104 03020101
300D0609 2A864886 F70D0101 05050003 41007F2E 7D1676B5 560EDD1E D80B4205
3B39A742 B8E06813 786E9992 2E4C5860 AB4AF193 63A34170 50BD756A AEA7086A
7A9FC2AC D1D8A3A0 CC6779D0 76CE7D1A 7F4C
quit
!
certificate 1FFC97F0000100000029
308204B3 3082045D A0030201 02020A1F FC97F000 01000000 29300D06 092A8648
86F70D01 01050500 3081AC31 21301F06 092A8648 86F70D01 09011612 64696E67
6A734073 7461722D 6E65742E 636E310B 30090603 55040613 02434E31 0F300D06
03550408 13064675 4A69616E 310F300D 06035504 07130646 755A686F 75312030
1E060355 040A1317 52656769 616E7420 4E657477 6F726B20 436F2E20 4C746431
1D301B06 0355040B 13145265 73656172 63682041 70617274 6D656E74 20353117
30150603 55040313 0E434120 74657374 20736572 76657230 1E170D30 35303332
31303632 3335395A 170D3036 30333231 30363333 35395A30 81A63122 30200609
2A864886 F70D0109 0116137A 68616F6A 756E4073 7461722D 6E65742E 636E310B
30090603 55040613 02434E31 0F300D06 03550408 13064675 4A69616E 310F300D
06035504 07130646 755A686F 75312030 1E060355 040A1317 52656769 616E7420
4E657477 6F726B20 436F2E20 4C746431 1D301B06 0355040B 13145265 73656172
63682041 70617274 6D656E74 20353110 300E0603 55040313 077A6861 6F6A756E
305C300D 06092A86 4886F70D 01010105 00034B00 30480241 00BB70FF 351D18F5
7735FE4F C890AF42 8E8744BA D946C4B8 61F046DF 614E4A37 D8A3BA80 7003D7E1
BC5394F3 58DDE033 4ABA82D1 AEAD4C10 3135C2AE BB58FA2F 75020301 0001A382
02633082 025F300E 0603551D 0F0101FF 04040302 06C03013 0603551D 25040C30
0A06082B 06010505 08020230 1D060355 1D0E0416 0414420E 87E0F7C1 B744AFE3
2C1EFD64 E03E2144 844C3081 E8060355 1D230481 E03081DD 80147243 841C2B03
452D8258 F5844377 F521AA4B 2C21A181 B2A481AF 3081AC31 21301F06 092A8648
86F70D01 09011612 64696E67 6A734073 7461722D 6E65742E 636E310B 30090603
55040613 02434E31 0F300D06 03550408 13064675 4A69616E 310F300D 06035504
07130646 755A686F 75312030 1E060355 040A1317 52656769 616E7420 4E657477
6F726B20 436F2E20 4C746431 1D301B06 0355040B 13145265 73656172 63682041
70617274 6D656E74 20353117 30150603 55040313 0E434120 74657374 20736572
76657282 1056AE07 3C10A17E 8C45AE8D 3F155233 57307F06 03551D1F 04783076
```

```
 3038A036 A0348632 68747470 3A2F2F7A 6A2D726F 75746572 2F436572 74456E72
 6F6C6C2F 43412532 30746573 74253230 73657276 65722E63 726C303A A038A036
 86346669 6C653A2F 2F5C5C7A 6A2D726F 75746572 5C436572 74456E72 6F6C6C5C
 43412532 30746573 74253230 73657276 65722E63 726C3081 AC06082B 06010505
 07010104 819F3081 9C304B06 082B0601 05050730 02863F68 7474703A 2F2F7A6A
 2D726F75 7465722F 43657274 456E726F 6C6C2F7A 6A2D726F 75746572 5F434125
 32307465 73742532 30736572 76657228 31292E63 7274304D 06082B06 01050507
 30028641 66696C65 3A2F2F5C 5C7A6A2D 726F7574 65725C43 65727445 6E726F6C
 6C5C7A6A 2D726F75 7465725F 43412532 30746573 74253230 73657276 65722831
 292E6372 74300D06 092A8648 86F70D01 01050500 034100AF 173B4A23 E95C8042
 ED4F2F97 0D869C1E 715800E6 F64F505F 1A6F291C 4B8C95C8 2FE04F9C CA81778F
 07A2DE20 C9640A8B DD36BCC0 359C26BB D5A5E434 B5F46B
 quit
 !
!
```

# The IKE negotiation uses a certificate for authentication by default

```
!
```

# Configure a transform set

```
crypto ipsec transform-set myset  esp-des
```

# Configure a dynamic crypto map set

```
crypto dynamic-map dy 1
 set transform-set myset
!
!
crypto map mymap 1 ipsec-isakmp dynamic dy
!
interface FastEthernet 0/0
 ip address 63.23.12.212 255.255.255.248
 crypto map mymap
 duplex auto
 speed auto
!
interface FastEthernet 0/1
 ip address 192.168.216.1 255.255.255.0
 ip address 192.168.217.1 255.255.255.0 secondary
 ip address 192.168.218.1 255.255.255.0 secondary
 ip address 192.168.219.1 255.255.255.0 secondary
 duplex auto
 speed auto
!
interface Null 0
!
```

```
interface Virtual-Template 1
 mtu 1400
 ip ospf mtu-ignore
 ip unnumbered FastEthernet 0/1
 ip mtu 1360
!
!
router ospf
 redistribute static subnets route-map static-ospf
 network 192.168.216.0 0.0.0.255 area 0.0.0.0
 network 192.168.217.0 0.0.0.255 area 0.0.0.0
 network 192.168.218.0 0.0.0.255 area 0.0.0.0
 network 192.168.219.0 0.0.0.255 area 0.0.0.0
!
line con 0
 exec-timeout 0 0
line aux 0
 disconnect-character 240
line vty 0 4
 exec-timeout 0 0
 privilege level 15
 no login
!
!
end
```

Configuration of router A:

```
hostname routerA
!
# Configure an access list
access-list 101 permit ip interface dialer 0 host 63.23.12.212
access-list 177 deny icmp any any unreachable
access-list 177 permit ip any any
dialer-list 1 protocol ip permit
!
l2tp-class l2x
 hostname rg36_1
!
pseudowire-class pw
 encapsulation l2tpv2
 protocol l2tpv2 l2x
 ip local interface dialer 0
!
!
```

Configure the certificate and root certificate for the local router

```
crypto pki certificate chain
 certificate ca 56AE073C10A17E8C45AE8D3F15523357
 3082032E 308202D8 A0030201 02021056 AE073C10 A17E8C45 AE8D3F15 52335730
 0D06092A 864886F7 0D010105 05003081 AC312130 1F06092A 864886F7 0D010901
 16126469 6E676A73 40737461 722D6E65 742E636E 310B3009 06035504 06130243
 4E310F30 0D060355 04081306 46754A69 616E310F 300D0603 55040713 0646755A
 686F7531 20301E06 0355040A 13175265 6769616E 74204E65 74776F72 6B20436F
 2E204C74 64311D30 1B060355 040B1314 52657365 61726368 20417061 72746D65
 6E742035 31173015 06035504 03130E43 41207465 73742073 65727665 72301E17
 0D303530 32323530 38343630 325A170D 30373033 30313032 33363233 5A3081AC
 3121301F 06092A86 4886F70D 01090116 1264696E 676A7340 73746172 2D6E6574
 2E636E31 0B300906 03550406 1302434E 310F300D 06035504 08130646 754A6961
 6E310F30 0D060355 04071306 46755A68 6F753120 301E0603 55040A13 17526567
 69616E74 204E6574 776F726B 20436F2E 204C7464 311D301B 06035504 0B131452
 65736561 72636820 41706172 746D656E 74203531 17301506 03550403 130E4341
 20746573 74207365 72766572 305C300D 06092A86 4886F70D 01010105 00034B00
 30480241 00D91F1A C60EF951 924CDC96 4D4443FA DABE53F2 DDF513B6 34B5A6A1
 9FFD57A1 6C6F7AF4 A113C159 3D0C4C3E 8E62DE76 D8A24CF2 2CF8DA82 AA17D3E8
 CC80C295 8F020301 0001A381 D33081D0 300B0603 551D0F04 04030201 C6300F06
 03551D13 0101FF04 05300301 01FF301D 0603551D 0E041604 14724384 1C2B0345
 2D8258F5 844377F5 21AA4B2C 21307F06 03551D1F 04783076 3038A036 A0348632
 68747470 3A2F2F7A 6A2D726F 75746572 2F436572 74456E72 6F6C6C2F 43412532
 30746573 74253230 73657276 65722E63 726C303A A038A036 86346669 6C653A2F
 2F5C5C7A 6A2D726F 75746572 5C436572 74456E72 6F6C6C5C 43412532 30746573
 74253230 73657276 65722E63 726C3010 06092B06 01040182 37150104 03020101
 300D0609 2A864886 F70D0101 05050003 41007F2E 7D1676B5 560EDD1E D80B4205
 3B39A742 B8E06813 786E9992 2E4C5860 AB4AF193 63A34170 50BD756A AEA7086A
 7A9FC2AC D1D8A3A0 CC6779D0 76CE7D1A 7F4C
 quit
 !
 certificate 11F7C51700010000003E
 308204B1 3082045B A0030201 02020A11 F7C51700 01000000 3E300D06 092A8648
 86F70D01 01050500 3081AC31 21301F06 092A8648 86F70D01 09011612 64696E67
 6A734073 7461722D 6E65742E 636E310B 30090603 55040613 02434E31 0F300D06
 03550408 13064675 4A69616E 310F300D 06035504 07130646 755A686F 75312030
 1E060355 040A1317 52656769 616E7420 4E657477 6F726B20 436F2E20 4C746431
 1D301B06 0355040B 13145265 73656172 63682041 70617274 6D656E74 20353117
 30150603 55040313 0E434120 74657374 20736572 76657230 1E170D30 35303431
 32303932 3935335A 170D3036 30343132 30393339 35335A30 81A43121 301F0609
 2A864886 F70D0109 01161264 696E676A 73407374 61722D6E 65742E63 6E310B30
 09060355 04061302 434E310F 300D0603 55040813 0646754A 69616E31 0F300D06
 03550407 13064675 5A686F75 3120301E 06035504 0A131752 65676961 6E74204E
 6574776F 726B2043 6F2E204C 7464311D 301B0603 55040B13 14526573 65617263
 68204170 6172746D 656E7420 35310F30 0D060355 04031306 64696E67 6A73305C
 300D0609 2A864886 F70D0101 01050003 4B003048 024100CD 2C3B2981 FF9BF7E6
 F9DFFCF9 495FE6AA 6691FD76 BB5EBEC3 5A1E48F8 8B75DD68 9E79AC5A 36C0B4F4
 AA959323 49EEEB7F 24B546B8 74421F17 401033AE EC3F4102 03010001 A3820263
 3082025F 300E0603 551D0F01 01FF0404 030204F0 30130603 551D2504 0C300A06
 082B0601 05050802 02301D06 03551D0E 04160414 216567F3 E72263B2 4990E14D
 ECC22471 1596A71F 3081E806 03551D23 0481E030 81DD8014 7243841C 2B03452D
 8258F584 4377F521 AA4B2C21 A181B2A4 81AF3081 AC312130 1F06092A 864886F7
 0D010901 16126469 6E676A73 40737461 722D6E65 742E636E 310B3009 06035504
 06130243 4E310F30 0D060355 04081306 46754A69 616E310F 300D0603 55040713
 0646755A 686F7531 20301E06 0355040A 13175265 6769616E 74204E65 74776F72
 6B20436F 2E204C74 64311D30 1B060355 040B1314 52657365 61726368 20417061
 72746D65 6E742035 31173015 06035504 03130E43 41207465 73742073 65727665
 72821056 AE073C10 A17E8C45 AE8D3F15 52335730 7F060355 1D1F0478 30763038
 A036A034 86326874 74703A2F 2F7A6A2D 726F7574 65722F43 65727445 6E726F6C
 6C2F4341 25323074 65737425 32307365 72766572 2E63726C 303AA038 A0368634
 66696C65 3A2F2F5C 5C7A6A2D 726F7574 65725C43 65727445 6E726F6C 6C5C4341
 25323074 65737425 32307365 72766572 2E63726C 3081AC06 082B0601 05050701
 0104819F 30819C30 4B06082B 06010505 07300286 3F687474 703A2F2F 7A6A2D72
 6F757465 722F4365 7274456E 726F6C6C 2F7A6A2D 726F7574 65725F43 41253230
 74657374 25323073 65727665 72283129 2E637274 304D0608 2B060105 05073002
 86416669 6C653A2F 2F5C5C7A 6A2D726F 75746572 5C436572 74456E72 6F6C6C5C
 7A6A2D72 6F757465 725F4341 25323074 65737425 32307365 72766572 2831292E
 63727430 0D06092A 864886F7 0D010105 05000341 00148479 89448BB7 E6D3A7A7
 34376464 C8D857C2 D9075263 9E278FC3 2D6C5041 036B66C7 F59ADCA5 39C9F824
 A40A4C57 05373965 66671538 921FF39C B95C90F4 3F
 quit
!
crypto pki revocation-check none
!
```

# Configure that DPD is triggered periodically

```
crypto isakmp keepalive 20 periodic
```

# Configure a transform set

```
crypto ipsec transform-set myset  esp-des
```

# Configure a crypto map set

```
crypto map mymap 1 ipsec-isakmp
 set peer 63.23.12.212
 set transform-set myset
 match address 101
!
!
!
interface FastEthernet 0/0
```

```
pppoe enable
  pppoe-client dial-pool-number 1 no-ddr
  duplex auto
  speed auto
!
interface FastEthernet 0/1
 ip address 192.168.161.1 255.255.255.0
 duplex auto
 speed auto
!
interface dialer 0
 mtu 1488
 encapsulation PPP
 ppp chap hostname abcd@163.com
 ppp chap password 0 123456
 ppp pap sent-username abcd@163.com password 0 123456
 ip access-group 177 in
 ip address negotiate
 crypto map mymap
 dialer pool 1
 dialer idle-timeout 1200
 dialer-group 1
 bandwidth 2048
!
interface Null 0
!
interface Virtual-ppp 1
 pseudowire 63.23.12.212 20 encapsulation l2tpv2 pw-class pw
 mtu 1400
 ppp pap sent-username RGOS password 0 RGOS
 ip ospf mtu-ignore
 no ip route-cache policy
 ip unnumbered FastEthernet 0/1
 ip mtu 1360
!
!
router ospf
 network 192.168.161.0 0.0.0.255 area 0.0.0.0
!
ip route 0.0.0.0 0.0.0.0 dialer 0
!
!
line con 0
line aux 0
line vty 0 4
```

```
 exec-timeout 0 0
 privilege level 15
 no login
!
!
end
```

For configurations of Router B and Router C, refer to the configuration of Router A.

Caution

Because the private key of a device is confidential information that does not exist in the system configuration file, when you copy and paste the preceding configuration information to the device console or use the **copy tft flash** command to copy this configuration file to the device configuration file **config.txt**, the preceding certificate-related configurations cannot be performed. To configure a certificate, you must run the **crypto pki import pem terminal** command to import a certificate. This example only shows the configurations that are visible to you.

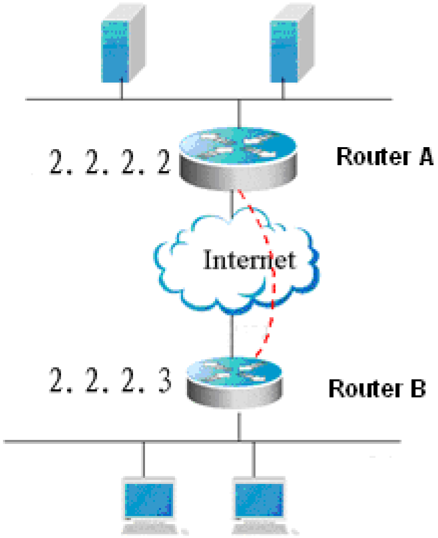
Reverse Route Injection

Analysis

In this case, the IP traffic between two subnets is protected by using the Router A as the central gateway at one side, and using Router B as the branch gateway at the other side, as shown in Figure 5. The following requirements should be met:

- The 3DES algorithm is used in phase 1.
- The tunnel mode is used.
- The protection method is ESP-3des (the encryption and authentication services are available).

Figure 13



Router configuration

10) Router A

```
ip host peerhost 2.2.2.2
ip access-list extended 110
10 permit ip host 2.2.2.3 host 2.2.2.2
crypto isakmp policy 1
authentication pre-share
!
!
crypto isakmp key 7 01334b46391e033004 address 0.0.0.0 0.0.0.0
crypto ipsec transform-set myset  esp-3des
crypto dynamic-map dymap 100
set transform-set myset
reverse-route
match address 111
!
!
crypto map mymap 7 ipsec-isakmp dynamic dymap
interface GigabitEthernet 1/0/0
 ip ref
ip address 2.2.2.3 255.255.255.0
crypto map mymap
duplex auto
speed auto
end
```

11) Router B

```
ip access-list extended 110
10 permit ip host 2.2.2.2 host 2.2.2.3
crypto isakmp policy 1
authentication pre-share
crypto isakmp key 7 0424100330052a1b15 address 0.0.0.0 0.0.0.0
crypto ipsec transform-set myset  esp-3des
crypto map mymap 10 ipsec-isakmp
set peer 2.2.2.3
set transform-set myset
match address 110
interface FastEthernet 0/0
ip address 2.2.2.2 255.255.255.0
crypto map mymap
duplex auto
speed auto
!
interface FastEthernet 0/1
ip address 200.1.1.10 255.255.255.0
duplex auto
speed auto
end
```

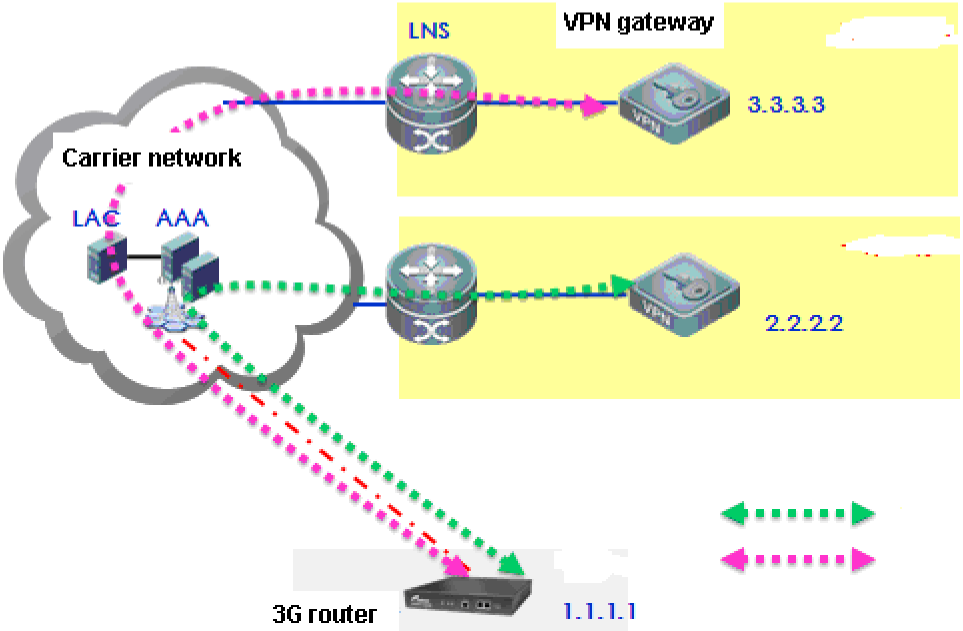
Mutual Backup of Multiple Peers

Analysis

In this case, there are multiple servers and each server uses a different certificate chain. The access device is configured with multiple peers and certificate chains. While the convergence device has the same configuration as that in the other case. As shown in the figure below, the following requirements should be met:

- Multiple VPN gateways are located in different places to provide access services.
- Every convergence router uses a different certificate chain.
- The access router changes according to the peer configuration.

Figure 14



Router configuration

12) Convergence router A

//Use the default policy.

```
!
crypto ipsec transform-set myset  esp-3des
crypto dynamic-map dymap 100
 set transform-set myset
 reverse-route
!
!
crypto map mymap 7 ipsec-isakmp dynamic dymap
interface GigabitEthernet 1/0/0
 ip ref
 ip address 2.2.2.3 255.255.255.0
 crypto map mymap
 duplex auto
 speed auto
end
```

13) Convergence router B whose configuration is the same as router A except the local address and certificate

//Use the default policy.

```
!
crypto ipsec transform-set myset  esp-3des
crypto dynamic-map dymap 100
 set transform-set myset
 reverse-route
!
!
crypto map mymap 7 ipsec-isakmp dynamic dymap
interface GigabitEthernet 1/0/0
 ip ref
 ip address 2.2.2.2 255.255.255.0
 crypto map mymap
 duplex auto
 speed auto
end
```

14) Access router A

```
ip access-list extended 110
 10 permit ip host 2.2.2.1 any
!
//Use the default policy: certificate-based authentication
crypto isakmp keepalive 10 2 periodic
//Use 3DES for encryption
crypto ipsec transform-set myset  esp-3des
!
crypto map mymap 7 ipsec-isakmp
set peer 2.2.2.2 //The default certificate chain is used if this parameter is not specified.
set peer 2.2.2.3 trustpoint backup
set transform-set myset
match address 100

interface GigabitEthernet 1/0/0
 ip ref
 ip address 2.2.2.1 255.255.255.0
 crypto map mymap
 duplex auto
 speed auto
end
```

Applying Profile crypto map entries in different tunnels

- Analysis

In this case, the IP traffic between two subnets is protected by using a Cisco device connected to Subnet A as the gateway at one side, and using a Ruijie device connected to Subnet B as the gateway at the other side. The following requirements should be met:

- The 3DES algorithm is used in phase 1
- The transmission mode is used.
- The protection method is ESP-DES-MD5 (the encryption and authentication services are available).

In the following application, a Cisco device is used as the center, and Ruijie devices are used as remote branches. See Figure 15.

Figure 15

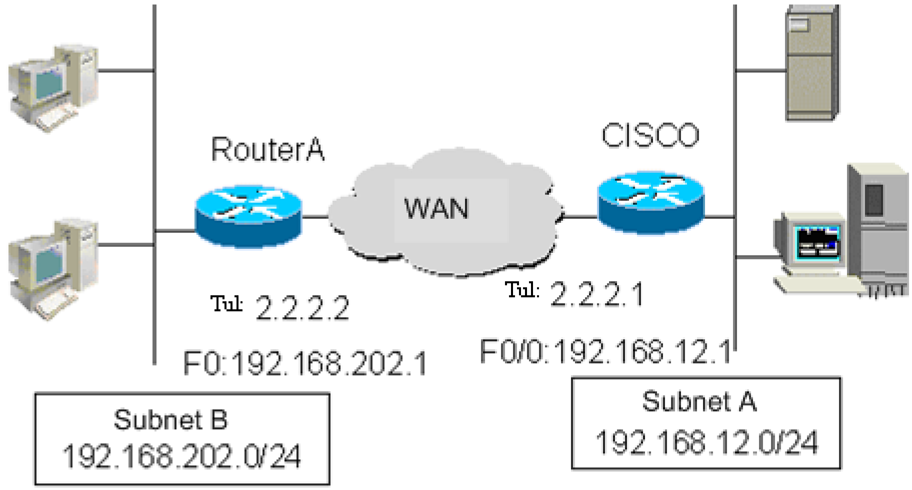
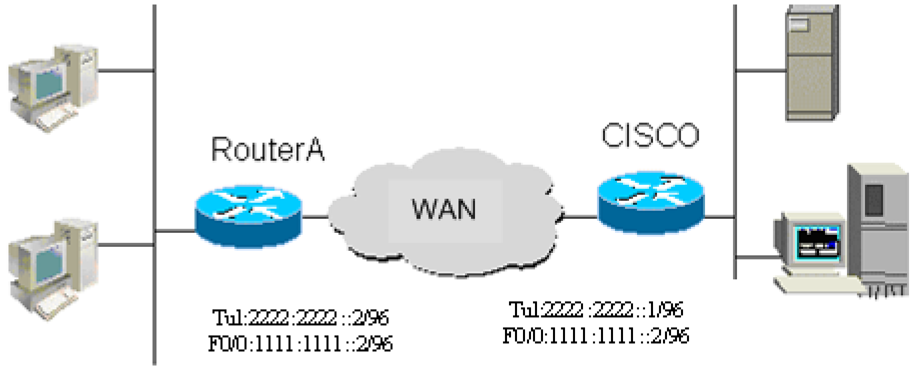


Figure 16



Applying the Profile map to a GRE tunnel

Configuration of Router A:

```
!
hostname "RouterA"
```

# Enable IKE

```
crypto isakmp enable
crypto isakmp policy 1
authentication pre-share
encrytiong 3des
```

# Configure a pre-shared key and a transform set

```
crypto isakmp key 0 123 address 192.168.12.1
crypto ipsec transform-set t1  esp-des esp-md5-hmac
mode transport
```

# Define a crypto map set

```
crypto ipsec profile profi-map
set transform-set t1
!
interface FastEthernet0
ip address 192.168.202.1 255.255.255.0
# Apply the crypto map to a GRE tunnel interface
interface tunnel 1
tunnel source  192.168.202.1
tunnel destination 192.168.12.1
tunnel protection ipsec profile profile-map
ip address 2.2.2.2 255.255.255.0
!
ip route 0.0.0.0 0.0.0.0 tunnel 1
```

Configuration of Cisco device:

```
!
hostname Cisco
```

# Define an IKE policy, using the pre-shared key for authentication, and using the default values for other parameters

```
crypto isakmp policy 1
authentication pre-share
encryption 3des
```

# Configure the pre-shared key

```
crypto isakmp key 0 123 address 192.168.202.1
```

# Define a transform set

```
crypto ipsec transform-set t1 esp-des esp-md5-hmac
```

```
mode transport
```

# Define a crypto map

```
crypto ipsec profile profile-name
set transform-set t1
!
interface FastEthernet0/0
ip address 192.168.12.1 255.255.255.0
```

# Apply the crypto map to the tunnel interface

```
interface tunnel 1
tunnel source  192.168.12.1
tunnel destination 192.168.202.1
tunnel protection ipsec profile profile-map
ip address 2.2.2.1 255.255.255.0
!
```

Applying the Profile map to an IPSEC-IPV4 tunnel

Configuration of Router A:

```
!
hostname "RouterA"
```

# Enable IKE

```
crypto isakmp enable
crypto isakmp policy 1
authentication pre-share
encrytiong 3des
```

# Configure a pre-shared key and a transform set

```
crypto isakmp key 0 123 address 192.168.12.1
crypto ipsec transform-set t1  esp-des esp-md5-hmac
mode transport
```

# Define a crypto map set

```
crypto ipsec profile profi-map
set transform-set t1
match any
!
interface FastEthernet0
ip address 192.168.202.1 255.255.255.0
# Apply the crypto map to the tunnel interface
interface tunnel 1
  tunnel mode ipip
tunnel source  192.168.202.1
tunnel destination 192.168.12.1
tunnel protection ipsec profile profile-map
ip address 2.2.2.2 255.255.255.0
!
ip route 0.0.0.0 0.0.0.0 tunnel 1

Configuration of Cisco device:
!
hostname Cisco
```

# Define an IKE policy, using the pre-shared key for authentication, and using the default values for other parameters

```
crypto isakmp policy 1
authentication pre-share
encryption 3des
```

# Configure a pre-shared key

```
crypto isakmp key 0 123 address 192.168.202.1
```

# Define a transform set

```
crypto ipsec transform-set t1 esp-des esp-md5-hmac
mode transport
```

# Define a crypto map

```
crypto ipsec profile profile-name
set transform-set t1
!
interface FastEthernet0/0
ip address 192.168.12.1 255.255.255.0
```

# Apply the crypto map to the tunnel interface

```
interface tunnel 1
tunnel mode ipsec ipv4
tunnel source  192.168.12.1
tunnel destination 192.168.202.1
tunnel protection ipsec profile profile-map
ip address 2.2.2.1 255.255.255.0
```

Applying the Profile map to an IPV6 tunnel

Configuration of Router A:

```
!
hostname "RouterA"
```

# Enable IKE

```
crypto isakmp enable
crypto isakmp policy 1
authentication pre-share
```

```
encrytiong 3des
```

# Configure a pre-shared key and a transform set

```
crypto isakmp key 0 123 ipv6  ::/0
crypto ipsec transform-set t1  esp-des esp-md5-hmac
mode transport
```

# Define a crypto map set

```
crypto ipsec profile profi-map
set transform-set t1
match any
!
interface FastEthernet0
ipv6 address 1111:1111::2/96
# Apply the crypto map to the tunnel interface
interface tunnel 1
  tunnel mode ipv6
tunnel source  1111:1111::2
tunnel destination 1111:1111::1
tunnel protection ipsec profile profile-map
ipv6 address 2222:2222::2/96
!
ipv6 route ::/0 tunnel 1
```

Configuration of Cisco device:

```
!
hostname Cisco
```

# Define an IKE policy, using the pre-shared key for authentication, and using the default values for other parameters

```
crypto isakmp policy 1
authentication pre-share
encryption 3des
# Configure a pre-shared key
crypto isakmp key 0 123 ipv6  ::/0
crypto ipsec transform-set t1  esp-des esp-md5-hmac
mode transport
```

# Define a crypto map set

```
crypto ipsec profile profi-map
set transform-set t1
!
interface FastEthernet0
ipv6 address 1111:1111::1/96
# Apply the crypto map to the tunnel interface
```

```
interface tunnel 1
  tunnel mode ipv6
tunnel source  1111:1111::1
tunnel destination 1111:1111::2
tunnel protection ipsec profile profile-map
ipv6 address 2222:2222::1/96
```

Applying the Profile map to an IPIP tunnel when NAT is available

Configuration of Router A:

```
!
hostname "RouterA"
```

# Enable IKE

```
crypto isakmp enable
crypto isakmp policy 1
authentication pre-share
encrytiong 3des
```

# Configure a pre-shared key and a transform set

```
crypto isakmp key 0 123 address 192.168.12.1
crypto ipsec transform-set t1  esp-des esp-md5-hmac
mode transport
```

# Define a crypto map set

```
crypto ipsec profile profi-map
set transform-set t1
!
interface FastEthernet0
ip address 192.168.202.1 255.255.255.0
```

# Apply the crypto map to the GRE tunnel interface

```
interface tunnel 1
   tunnel mode ipip
tunnel source  192.168.202.1
tunnel destination 192.168.12.1
tunnel protection ipsec profile profile-map
ip address 2.2.2.2 255.255.255.0
!
ip route 0.0.0.0 0.0.0.0 tunnel 1
```

Configuration of the NAT device:

```
hostname "NAT"
```

#An ACL for NAT translation

```
ip access standard 1
10 permit any
!
```

#Connected to Router A

```
interface FastEthernet 0/0
ip address 192.168.202.2 255.255.255.0
ip nat inside
!
```

#Connected to Router B

```
interface FastEthernet0/1
ip add 192.168.12.2 255.255.255.0
ip nat outside
!
```

#NAT translation rule, translating the source IP address of packets through F0/0 into the IP address of F0/1

```
ip nat inside source list 1 interface fastEthernet 0/1
!
```

Configuration of Cisco device:

```
!
hostname Cisco
```

# Define an IKE policy, using the pre-shared key for authentication, and using the default values for other parameters

```
crypto isakmp policy 1
authentication pre-share
encryption 3des
```

# Configure a pre-shared key

```
crypto isakmp key 0 123 address 192.168.202.1
```

# Define a transform set

```
crypto ipsec transform-set t1 esp-des esp-md5-hmac
mode transport
```

# Define a crypto map

```
crypto ipsec profile profile-name
set transform-set t1
!
interface FastEthernet0/0
ip address 192.168.12.1 255.255.255.0
```

# Apply the crypto map to the tunnel interface

```
interface tunnel 1
  tunnel mode ipip
tunnel source  192.168.12.1
tunnel destination 192.168.12.2
tunnel protection ipsec profile profile-map
ip address 2.2.2.1 255.255.255.0
!
```

Applying the Profle map to an IPSEC-Ipv4 tunnel when NAT is available

Configuration of Router A:

```
!
hostname "RouterA"
```

# Enable IKE

```
crypto isakmp enable
crypto isakmp policy 1
authentication pre-share
encrytiong 3des
```

# Configure a pre-shared key and a transform set

```
crypto isakmp key 0 123 address 192.168.12.1
crypto ipsec transform-set t1  esp-des esp-md5-hmac
mode transport
```

# Define a crypto map set

```
crypto ipsec profile profi-map
set transform-set t1
match any
!
interface FastEthernet0
ip address 192.168.202.1 255.255.255.0
```

# Apply the crypto map to the GRE tunnel interface

```
interface tunnel 1
  tunnel mode ipip
tunnel source  192.168.202.1
tunnel destination 192.168.12.1
tunnel protection ipsec profile profile-map
ip address 2.2.2.2 255.255.255.0
!
ip route 0.0.0.0 0.0.0.0 tunnel 1
```

Configuration of the NAT device:

```
hostname "NAT"
```

#An ACL for NAT translation

```
ip access standard 1
10 permit any
!
```

#Connected to Router A

```
interface FastEthernet 0/0
ip address 192.168.202.2 255.255.255.0
ip nat inside
!
```

#Connected to Router B

```
interface FastEthernet0/1
ip add 192.168.12.2 255.255.255.0
ip nat outside
!
```

#NAT translation rule, translating the source IP address of packets through F0/0 into the IP address of F0/1

```
ip nat inside source list 1 interface fastEthernet 0/1
!
```

Configuration of Cisco device:

```
!
hostname Cisco
```

# Define an IKE policy, using the pre-shared key as the authentication method, and using the default values for other parameters

```
crypto isakmp policy 1
authentication pre-share
encryption 3des
```

# Configure a pre-shared key

```
crypto isakmp key 0 123 address 192.168.202.1
```

# Define a transform set

```
crypto ipsec transform-set t1 esp-des esp-md5-hmac
mode transport
```

# Define a crypto map

```
crypto ipsec profile profile-name
set transform-set t1
!
```

```
interface FastEthernet0/0
ip address 192.168.12.1 255.255.255.0
```

# Apply the crypto map to the tunnel interface

```
interface tunnel 1
tunnel mode ipsec ipv4
tunnel source  192.168.12.1
tunnel destination 192.168.12.2
tunnel protection ipsec profile profile-map
ip address 2.2.2.1 255.255.255.0
!
```

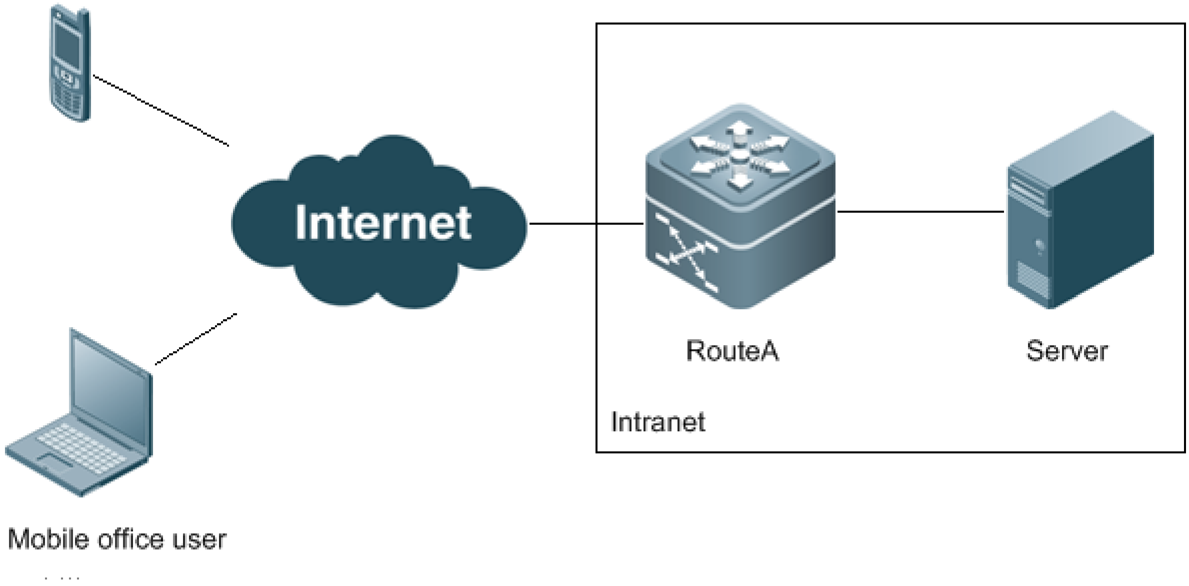
Extended authentication

- Analysis

In this case, a mobile office user wants to establish an IPSec connection with Route A through Internet for access to intranet resources, as shown in Figure 9. The following requirements should be met:

- Authentication is required.
- IPSec policies are dynamically downloaded to clients.

Figure 17



Configuration of Route A:

# Configure AAA authentication

```
aaa new-model
```

# Configure local authentication

```
aaa authentication login lab-remote-access local
```

# Configure the username and password for authentication

```
username ruijie password 0 ruijie
```

# Configure an IKE policy

```
crypto isakmp policy 10
 encr 3des
 authentication pre-share
 group 2
```

# Configure an IKE address pool

```
crypto isakmp ippool Remote-Pool
 address 172.16.1.200 172.16.1.250
```

# Configure a policy to be delivered to clients

```
crypto isakmp client configuration group test
key VPNKEY
dns 220.170.0.18
pool Remote-Pool
network center 192.168.52.0/24
```

# Configure a transform set

```
crypto ipsec transform-set VPNTRANSFORM esp-3des esp-sha-hmac
```

# Define a dynamic crypto map set

```
crypto dynamic-map Dynamic-Map 10
set transform-set VPNTRANSFORM
reverse-route
```

# Use AAA authentication

```
crypto map ClientMap client authentication list lab-remote-access
```

# Add a dynamic crypto map set to the static crypto map set

```
crypto map ClientMap 65535 ipsec-isakmp dynamic Dynamic-Map
```

# Apply the crypto map to the interface

```
interface FastEthernet0/0
ip address 61.168.202.1 255.255.255.0
crypto map ClientMap
!
interface FastEthernet0/1
ip address 192.168.202.1 255.255.255.0
```

IPSEC OVER MPLS

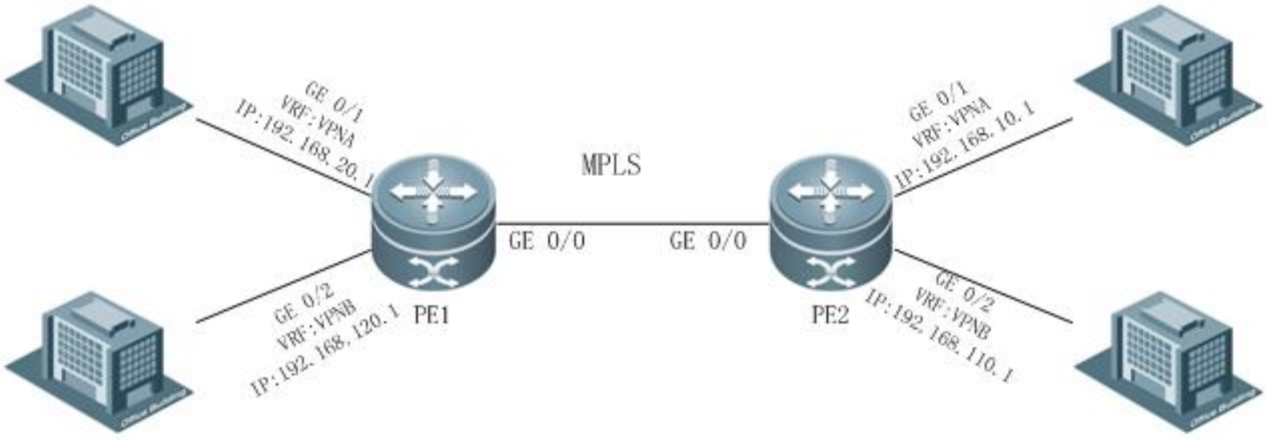
Analysis

MPLS on E-government extranet connects networks of official organs. To enhance MPLS networking security, encryption protection is adopted for network communication data. The following requirements should be met:

The data transferring on MPLS networks must be encrypted by IPSec.

To make sure the IPSEC encryption do not interfere with data transferring on MPLS networks, adopt IPSEC OVER MPLS.

Figure 18



PE1：

# configure VRF

```
ip vrf VPNA
 rd 1:100
 route-target both 1:100
!
ip vrf VPNB
 rd 1:200
 route-target both 1:200
```

# configure MPLS network

```
mpls ip
interface Loopback 0
 ip address 172.168.0.1 255.255.255.255
router bgp 1
 bgp log-neighbor-changes
 neighbor 172.168.0.2 remote-as 1
 neighbor 172.168.0.2 update-source Loopback 0
 !
 address-family ipv4
  neighbor 172.168.0.2 activate
 exit-address-family
 !
 address-family vpnv4 unicast
  neighbor 172.168.0.2 activate
  neighbor 172.168.0.2 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf VPNA
  maximum-prefix 10000
  redistribute connected
  neighbor 172.168.10.2 remote-as 65002
  neighbor 172.168.10.2 activate
 exit-address-family
 !
 address-family ipv4 vrf VPNB
  maximum-prefix 10000
  redistribute connected
  neighbor 172.168.10.2 remote-as 65002
  neighbor 172.168.10.2 activate
 exit-address-family
!
!
router ospf 10
 network 172.168.0.1 0.0.0.0 area 0
 network 172.168.10.0 0.0.0.255 area 0
 network 192.168.20.0 0.0.0.255 area 0
 network 192.168.120.0 0.0.0.255 area 0!
!
mpls router ldp
 ldp router-id interface Loopback 0 force
```

# configure IPSEC

```
ip access-list extended 110
 10 permit ip 192.168.20.0 0.0.0.255 192.168.10.0 0.0.0.255
!
!
ip access-list extended 111
 10 permit ip 192.168.120.0 0.0.0.255 192.168.110.0 0.0.0.255
!
crypto isakmp policy 1
 encryption 3des
 authentication pre-share
!
crypto ipsec transform-set myset esp-3des esp-sha-hmac
!
crypto map mymap1 1 ipsec-isakmp
 set local 192.168.20.2
 set peer 192.168.10.2
 set transform-set myset
 match vrf VPNA
 match address 110
!
crypto map mymap1 2 ipsec-isakmp
 set local 192.168.120.2
 set peer 192.168.110.2
 set transform-set myset
 match vrf VPNB
 match address 111
!
interface GigabitEthernet 0/0
 ip address 172.168.10.1 255.255.255.0
 label-switching
 mpls ip
 crypto map mymap
 duplex auto
 speed auto
!
interface GigabitEthernet 0/1
 ip vrf forwarding VPNA
 ip address 192.168.20.2 255.255.255.0
duplex auto
 speed auto
!
interface GigabitEthernet 0/1
 ip vrf forwarding VPNB
 ip address 192.168.120.2 255.255.255.0
duplex auto
 speed auto
```

PE2：

# configure VRF

```
ip vrf VPNA
 rd 1:100
 route-target both 1:100
!
ip vrf VPNB
 rd 1:200
 route-target both 1:200
```

# configure MPLS network

```
mpls ip
!
router bgp 1
```

```
 bgp log-neighbor-changes
 neighbor 172.168.0.1 remote-as 1
 neighbor 172.168.0.1 update-source Loopback 0
 !
 address-family ipv4
  neighbor 172.168.0.1 activate
 exit-address-family
 !
 address-family vpnv4 unicast
  neighbor 172.168.0.1 activate
  neighbor 172.168.0.1 send-community extended
 exit-address-family
 !
 address-family ipv4 vrf VPNA
  maximum-prefix 10000
  redistribute connected
  neighbor 172.168.10.1 remote-as 65002
  neighbor 172.168.10.1 activate
 exit-address-family
 !
 address-family ipv4 vrf VPNB
  maximum-prefix 10000
  redistribute connected
  neighbor 172.168.10.1 remote-as 65002
  neighbor 172.168.10.1 activate
 exit-address-family
!
!
!
!
router ospf 10
 network 172.168.0.2 0.0.0.0 area 0
 network 172.168.10.0 0.0.0.255 area 0
 network 192.168.10.0 0.0.0.255 area 0
 network 192.168.110.0 0.0.0.255 area 0
!
!
!
!
mpls router ldp
 ldp router-id interface Loopback 0 force
```

# configure IPSEC

```
crypto isakmp policy 1
 encryption 3des
 authentication pre-share
!
!
crypto isakmp key 7 021211644c536854774c address 0.0.0.0 0.0.0.0
crypto ipsec transform-set myset esp-3des esp-sha-hmac
!
crypto dynamic-map dy 1
 set transform-set myset
 reverse-route
!
crypto map mymap 1 ipsec-isakmp dynamic dy
!
interface GigabitEthernet 0/0
 ip address 172.168.10.1 255.255.255.0
 label-switching
 mpls ip
 crypto map mymap
 duplex auto
 speed auto
!
interface GigabitEthernet 0/1
 ip vrf forwarding VPNA
 ip address 192.168.10.2 255.255.255.0
duplex auto
 speed auto
!
interface GigabitEthernet 0/1
 ip vrf forwarding VPNB
 ip address 192.168.110.2 255.255.255.0
duplex auto
 speed auto
```

Configuring the Digital Certificate

Overview

To ensure the security, authenticity, reliability, integrity and non-repudiation of the information transmitted between clients over network, the identity of clients must be verified, while the digital certificate is one of the methods to realize this function. PKI digital certificate technology associates the identity of individual or entity with a public key, and centrally issues the certificate through Certificate Authority (CA) to guarantee the validity and security of certificates. The digital certificates are electronic files issued by CA and binding the identity, public key and CA signature of the entity, with public key and private key forming a key pair in the public key cryptography system. Both sides of communication verify the validity of the certificate through CA signature in digital certificate, and use the public key contained in the certificate to verify the digital signature created by the peer device using the private key, thus completing authentication. There are two types of digital certificates: X.509 certificates and PGP certificates. X.509 certificates are supported by Ruijie products.

Digital certificates can be used in the IKE negotiation and SSL of IPSec. Certificate authentication can be used in the IKE configuration of IPSec and boasts the following merits:

- Higher security than PSK
- No need to separately configure PSK between every two peers of communication (easy to use)
- Security problems caused by key compromise can be addressed through certificate revocation.
- Use of overdue keys can be avoided by controlling the duration of key pairs through certificate validity

The X.509 certificate on the router can be acquired manually and through SCEP protocol. The merits of using the SCEP protocol to acquire the digital certificate of the router are shown below:

- The private key remains in the cryptographic equipment, ensuring higher security.
- The SCEP protocol adopts PKCS7 digital envelop during communication, ensuring the security of communication process.
- Supported by CA, SCEP is capable of updating digital certificates automatically.

Terminology

Public-Key Infrastructure (PKI) is a set of hardware, software, people, policies, and procedures needed to create, manage, distribute, use, store, and revoke digital certificates.

X.509: X.509 is an international standard recommended by ITU-T, and defines a widely accepted PKI, including data format and the process of public key distribution through the digital certificate issued by CA.

CA: As an authoritative, trustworthy and impartial third-party organization, CA is responsible for issuing and managing the digital certificates of all entities participating in online transaction. It effectively manages the key and issues digital certificates to prove the validity of such key, and associates the public key with one entity.

Root CA: CA at the top of the hierarchy.

Certificate or digital certificate: In this chapter, it refers to X.509 certificate (data structure defined by X.509), and is used to associate an entity with a public key to indicate the identity of the entity. A certificate contains a public key, name and digital signature of CA. Generally, the certificate also contains the validity period of the key, name of CA, serial number and etc, with format complying with ITUT X.509 standard.

CA root certificate: the self-signed certificate issued by root CA for itself; it is used to sign the other certificates issued by root CA.

Privacy-enhanced Mail (PEM): base64 encoded text format defined in RFC 1421-RFC 1424; generally used for e-mail and certificate import/export.

PKCS: A group of public-key cryptography standards devised and published by RSA Security, and is a widely-applied industry standard in information transfer. PKCS#1 defines a RSA encryption and signature algorithm; PKCS#7 defines a syntactic representation of enciphered message; PKCS#12 defines a method to create the security archive (PKCS12 can contain certificate, private key and other security achieves, and is a commonly used format for issuing certificates). The files exported in PKCS format are DER encoded binary files, and sometimes need to be converted into PEM encoded text files.

Certificate Revocation List (CRL): a list with time stamp to specify the certificates revoked by a CA. It can be freely obtained from the public directory, and is one of the two methods specified by Internet Public-Key Infrastructure (X.509) working group (PKIX) to check certificate state. Each certificate in the CRL is identified using its serial number. Therefore, by querying the serial number of certificate in the recently released CRL, the user's certificate system can check whether a certificate has been revoked; if the certificate is contained in CRL, then the certificate should be rejected. X.509 version 2 CRL contains version number, issuer DN (globally unique), validity period, serial number of the revoked certificate, time of revocation, reason of revocation, the algorithm for CA to issue this CRL and the signature thereof.

Simple Certificate Enrollment Protocol (SCEP): A draft protocol defined by Cisco to securely apply for a certificate for the router from CA. It is currently deployed on various network devices.

Configuring CA Server and Applying for & Exporting a Certificate

Installing the Certificate Services on a Windows 2003 Server

Step 1: Select Add/Remove Programs in Control Panel and click Add/Remove Windows Components.

Step 2: Select **Certificate Services** in the pop-up window.

Step 3: Click **Next** and select **Stand-alone root CA** in the pop-up window.

Step 4: Click **Next** and fill in CA related information.

Step 5: Click **Next** and set a database directory (or use the default path).

Step 6: Click **Next** to install the certificate services.

Setting Up CA Server on IIS

Step 1: Select Add/Remove Programs and click Add/Remove Windows Components.

Step 2: Select **Application Server** and then click **Details**.

Step 3: Select **Internet information services (IIS)** and then click **Details**.

Step 4: Select **World Wide Web Service** and then click **Details**.

Step 5: In the following dialog box, select **Active Server Pages** and **World Wide Web Service** and click **OK** for three times. If **World Wide Web Publishing Service** is selected, **Common Files** and **Internet Service Manager** will be selected as well by default.

Step 6: Return to the main window and click **Next** to proceed with installation.

Step 7: Enter Control Panel and select Administrative Tools; select Internet Services Manager.

Step 8: Right-click the default website.

Step 9: Select **Virtual Directory**, as shown below.

Step 10: Click **Next** and enter the alias of the virtual directory.

Step 11: Click **Next** and configure a content directory.

Step 12: Click the **Browse** button and select the directory for installing certificate services.

Note

On a Windows 2003 server, the installation directory of certificate services is shown below:

Step 13: Click **Next** and complete virtual directory configuration using default settings.

Step 14: Start the Internet explorer on the CA server and type in "127.0.0.1/Cert_Server".

Applying for and Exporting a Certificate

Currently, certificates to be installed on Ruijie routers are acquired on the PC. Suppose the IP address of a CA server is 192.168.64.130. Perform the following steps to apply for a certificate.

Step 1: Start the Internet explorer on a client (usually a PC) and type in "192.168.64.130/Cert_Server".

Step 2: Click **Next** and select **Advanced Application** as the application type.

Step 3: Click **Next** and select a method of submitting an application (using a form).

Step 4: Click **Next** and fill in detailed information.

Note

Mark keys as exportable must be checked, as RSA key pairs applicable to Ruijie routers are generated by a CA server, and certificates and key pairs must be exported eventually.

Step 5: Click **Submit** to complete certificate application.

Step 6: Return to homepage and select **Check on a pending certificate**.

Step 7: Click **Next**. If the CA center has issued a certificate, the certificate is displayed.

Step 8: Click **Next** and prepare to install this certificate, as shown below:

Step 9: Click **Install this certificate**. The system will prompt that this certificate is successfully installed.

Step 10: After installation of a certificate, the certificate needs to be exported.

Step 11: Select **Internet Options** and click the **Content** tab.

Step 12: Click the **Certificate** button.

Step 13: Select the certificate to be exported and click **Export**:

Step 14: Click **Next**.

Step 15: Click **Next**.

Note

Include all certificates in the certification path if possible must be checked. This export certificate file contains the issued certificate, private key and CA root certificate.

Step 16: Click **Next** and enter a file protection password.

Step 17: Click **Next** and enter the name of the file saving this certificate.

Step 18: Click **Next** and confirm the information, and then Click **Finish** to export the certificate.

Step 19: The certificate is exported as a pfx file. Then you need to convert the pfx file into a pem file by using "pfx2pem" tool available in the CD-ROM.

Step 20: Click **Convert**,

By this time, you will find the converted pem file in the corresponding directory. Use the Wordpad program to open this pem file; you will find the private key, certificate and the corresponding root certificate.

Caution

The password of source file is the password entered while exporting the certificate on IE; the password of the target file is the password you need to set currently. This password must be entered when you import the certificate to a device. See the "Exporting a certificate" section.

Configuring the URL of CRL

After the certificate services are installed, the default URLs of CRL are:

http://%SERVER_DNS_NAME%/CertEnroll/%CA_NAME%%CRL_SUFFIX%.crl

file://\\%SERVER_DNS_NAME%\CertEnroll\%CA_NAME%%CRL_SUFFIX%.crl

Where, the default value of *%SERVER_DNS_NAME%* is the hostname of this system rather than the domain name or IP address; if the domain name or stationary IP address must be used, CA server configurations must be changed. To change CA server configurations, perform the following steps:

Step 1: Select **Control Panel** and double-click **Administrative Tools**, and select **Certification Authority**.

Step 2: Right-click **CA Server**.

Step 3: Select **Properties** and click the **Policy Module** tab in the pop-up window.

Step 4: Click **Configure** and click the **X.509 Extensions** tab in the **Properties** window.

Step 5: Click **Add CDP**.

Step 6: Click **OK** to add a new CRL address

Example:

To specify an IP address:

Type in: http://192.168.64.130/CertEnroll/%CA_NAME%%CRL_SUFFIX%.crl

Type in: file:\\192.168.64.130\CertEnroll\%CA_NAME%%CRL_SUFFIX%.crl

To specify a domain name:

Type in: http://www.ruijie.com.cn/CertEnroll/%CA_NAME%%CRL_SUFFIX%.crl

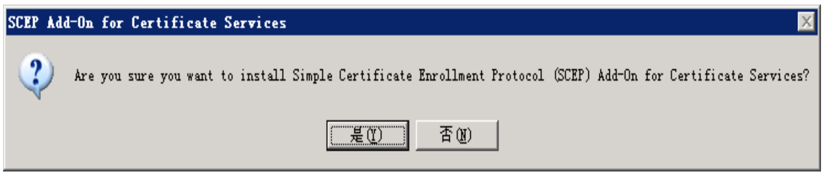
Type in: file://\\www.ruijie.com.cn\CertEnroll\%CA_NAME%%CRL_SUFFIX%.crl

Note

Currently, only the first CRL address is considered valid on Ruijie routers. Therefore, the desired CRL address should be placed on the top.

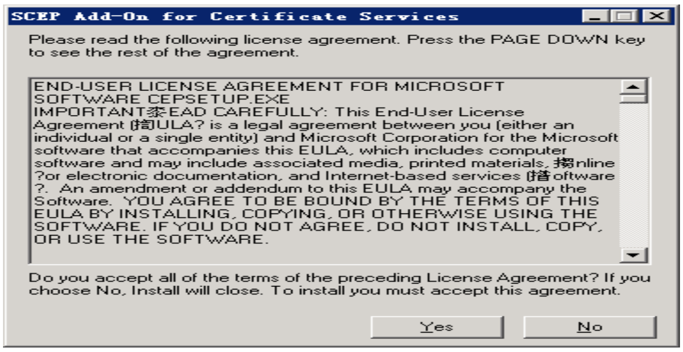
Installing SCEP Add-on

Step 1: Download the SCEP add-on for Windows Server 2003 from the following website: http://go.microsoft.com/fwlink/?LinkId=32060, and then double-click it to install.



Are you sure you want to install Simple Certificate Enrollment Protocol (SCEP) Add-On for Certificate Services?

Step 2: Click **Yes** to accept the license agreement.



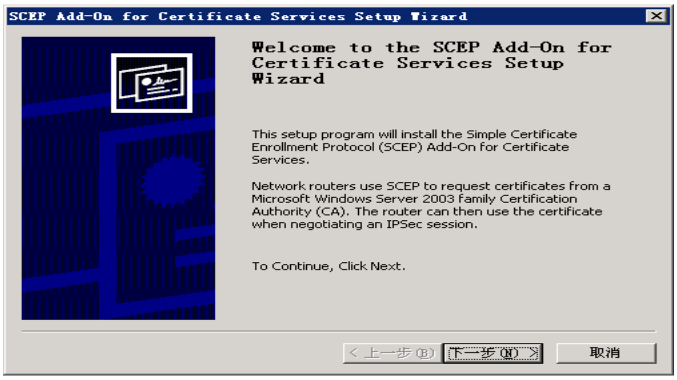
SCEP Add-On for Certificate Services

Please read the following license agreement. Press the PAGE DOWN key to see the rest of the agreement.

END-USER LICENSE AGREEMENT FOR MICROSOFT SOFTWARE CEPSETUP.EXE
IMPORTANT菍EAD CAREFULLY: This End-User License Agreement (挘ULA? is a legal agreement between you (either an individual or a single entity) and Microsoft Corporation for the Microsoft software that accompanies this EULA, which includes computer software and may include associated media, printed materials, 揂nline ?or electronic documentation, and Internet-based services (揝oftware ?. An amendment or addendum to this EULA may accompany the Software. YOU AGREE TO BE BOUND BY THE TERMS OF THIS EULA BY INSTALLING, COPYING, OR OTHERWISE USING THE SOFTWARE. IF YOU DO NOT AGREE, DO NOT INSTALL, COPY, OR USE THE SOFTWARE.

Do you accept all of the terms of the preceding License Agreement? If you choose No, Install will close. To install you must accept this agreement.

Step 3: Click **Next** to proceed with installation.



SCEP Add-On for Certificate Services Setup Wizard

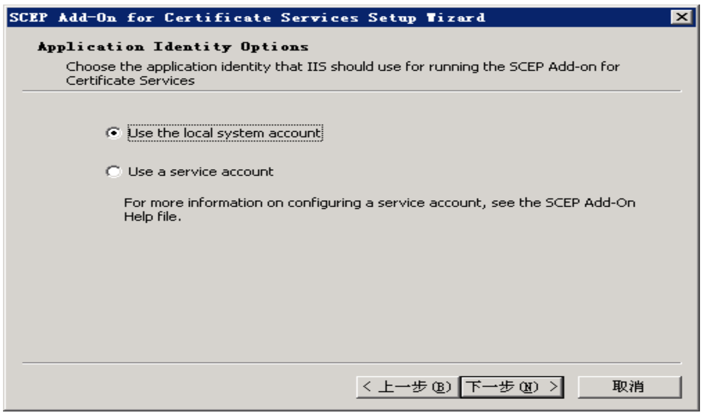
Welcome to the SCEP Add-On for Certificate Services Setup Wizard

This setup program will install the Simple Certificate Enrollment Protocol (SCEP) Add-On for Certificate Services.

Network routers use SCEP to request certificates from a Microsoft Windows Server 2003 family Certification Authority (CA). The router can then use the certificate when negotiating an IPSec session.

To Continue, Click Next.

Step 4: Select **Use the local system account** and then click **Next**.



SCEP Add-On for Certificate Services Setup Wizard

Application Identity Options

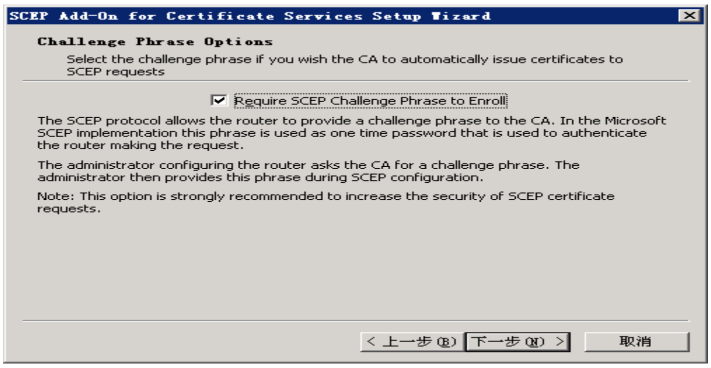
Choose the application identity that IIS should use for running the SCEP Add-on for Certificate Services

● Use the local system account

○ Use a service account

For more information on configuring a service account, see the SCEP Add-On Help file.

Step 5: It is suggested that you check **Require SCEP Challenge Phrase to Enroll**. After the device is ready to use CA to enroll, visit http://ca/certsrv/mscep/msdep.dll (from any client). The user will be required to provide the "passphrase" needed for enrollment. The "passphrase" is effective within 60 minutes. Click **Next**.



SCEP Add-On for Certificate Services Setup Wizard

Challenge Phrase Options

Select the challenge phrase if you wish the CA to automatically issue certificates to SCEP requests

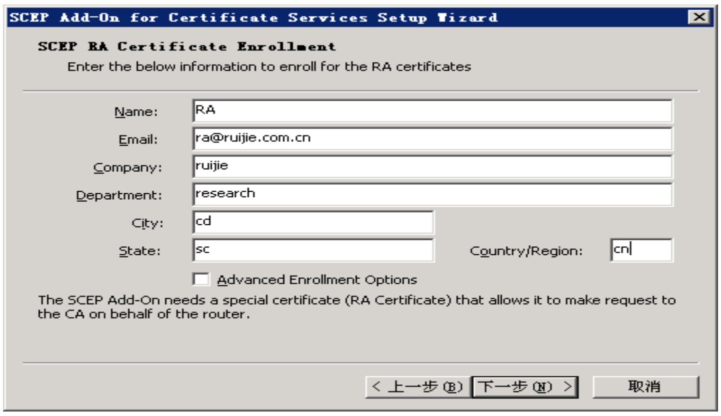
☑ Require SCEP Challenge Phrase to Enroll

The SCEP protocol allows the router to provide a challenge phrase to the CA. In the Microsoft SCEP implementation this phrase is used as one time password that is used to authenticate the router making the request.

The administrator configuring the router asks the CA for a challenge phrase. The administrator then provides this phrase during SCEP configuration.

Note: This option is strongly recommended to increase the security of SCEP certificate requests.

Step 6: Click **Next**.



SCEP Add-On for Certificate Services Setup Wizard

SCEP RA Certificate Enrollment

Enter the below information to enroll for the RA certificates

Name: RA

Email: ra@ruijie.com.cn

Company: ruijie

Department: research

City: cd

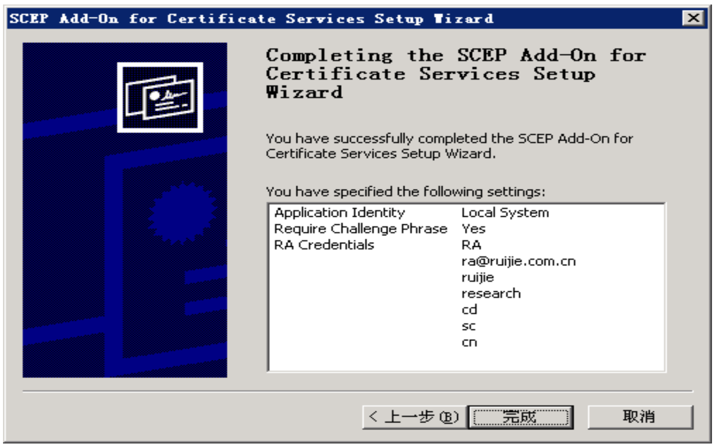
State: sc

Country/Region: cn

□ Advanced Enrollment Options

The SCEP Add-On needs a special certificate (RA Certificate) that allows it to make request to the CA on behalf of the router.

Step 7: Check the settings and click **Finish**.



SCEP Add-On for Certificate Services Setup Wizard

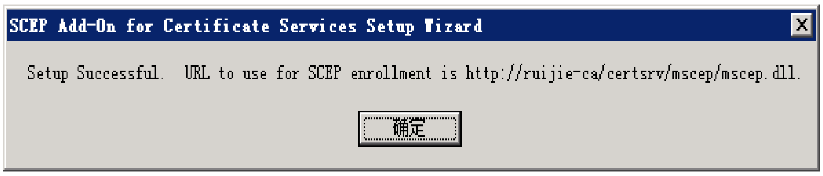
Completing the SCEP Add-On for Certificate Services Setup Wizard

You have successfully completed the SCEP Add-On for Certificate Services Setup Wizard.

You have specified the following settings:

```
Application Identity       Local System
Require Challenge Phrase   Yes
RA Credentials             RA
                           ra@ruijie.com.cn
                           ruijie
                           research
                           cd
                           sc
                           cn
```

Step 8: The URL used for SCEP enrollment is shown below:



SCEP Add-On for Certificate Services Setup Wizard

Setup Successful. URL to use for SCEP enrollment is http://ruijie-ca/certsrv/mscep/mscep.dll.

Digital Certificate Configuration

Digital Certificate Configuration Tasks

The purpose of digital certificate configuration is to enable the device to have its own certificate and verify the certificate of the communication peer.

Digital certificate configuration involves the following tasks:

- **Acquire certificate:** Acquire CA root certificate and the router certificate and private key issued directly by this CA. RGOS currently supports offline certificate application. The detailed steps are shown in the "Applying for and Exporting a Certificate" section. It also supports online SCEP certificate application and application for a certificate application file.

- **Import certificate:** Import the CA root certificate and router certificate and private key issued directly by this CA into the device.
- **Acquire router certificate through SCEP:** Acquire router certificate from the specified CA through SCEP.
- **Certificate configuration command (optional):** Configure a certificate chain and configure a certificate by importing binary certificate file (DER formatted file) in hexadecimal format.
- **Configure certificate revocation checking policy (optional):** Configure whether to check CRL during certificate check in order to verify whether the certificate has been revoked.
- **Download CRL (optional):** Configure the URL for downloading a CRL and start CRL download immediately through the command.

Importing a Certificate

Through the aforementioned steps, you may have acquired PEM-formatted CA root certificate, router certificate and private key. Run the following commands in global configuration mode to import these contents into the device:

Command	Function
Ruijie(config)# **crypto pki import pem terminal** *password* [**id** *string*]	Starts the interactive process to import CA root certificate, router certificate and private key. Password refers to the protective password of PEM-formatted private key. For details, see the "Applying for and Exporting a Certificate" section.

Caution The system will check certificate validity during the import process, and expired or inactive certificates will not be imported. Therefore, before importing a certificate, you must check whether the system time is the current Beijing time. To change system time, you can use the **clock** or **calendar** command. Refer to the "Basic System Management" chapter of *Basic Configuration Guide* for details. If the certificate is sm2, you need to set the ID. You can use the **crypto pki sm2-identity** command in global configuration mode to configure the ID.

Certificate import interface and process are shown below:

Assuming that PEM-formatted certificate data you acquired have been archived in one file, use a text editor (Windows Wordpad is recommended) to open the file and you will see the following contents:

```
Bag Attributes
localKeyID: 01 00 00 00
1.3.6.1.4.1.311.17.1: Microsoft Base Cryptographic Provider v1.0
friendlyName: 0929c7381e7517bdc65cdc7cc2ea0374_60e7aaa8-2e04-4953-9ba8-96bcaf0bfdf7
Key Attributes
X509v3 Key Usage: 10
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,251F9D955610C376
GDG2s1mbs/MJCpo5w2bu972jK1OZYtv3RQunH4I29c9H5uq3LtyvNA9RwrlpRQ3t
iUmkvQrU3/6SBp4Rqx1EU2UWgv1KRqqYwRVbdPdBZYVJLrso3Ov/9eaS4TiD+4Dl
NfJ1sAA4OONdVKDCLcGZIB43Wq5rAlqzsyjcF6tx3fWsSankVjQfroTv7UvP+ijj
```

```
uGndmJwbXEiATxlt+Smvt2/CGjr8nIC55T1W+tW0itkBdZhnvBJekOFM4BdgoLZc
3vueTIHmTurHvvdLIytYjQHsxVsf3vRGMcQhohM98nAYsIDBil4OIh1hc+ZnhGsn
TFLPMmMuJnBWMYopfaMPNrcdbpu+n4Qj2QiRoVTEoI7P1IAY/Oa2uc+kDuUX3KlW
sQQPnFNiU0Q/T9BrsolxI2Wkak7cvaNxbmhuU+5wNUGybQfcfP3CWg==
-----END RSA PRIVATE KEY-----
Bag Attributes
   localKeyID: 01 00 00 00
subject=/Email=dingjs@star-net.cn/C=CN/ST=FuJian/L=FuZhou/O=Regiant Network Co.
Ltd/OU=Research Apartment 5/CN=dingjs
issuer=/Email=dingjs@star-net.cn/C=CN/ST=FuJian/L=FuZhou/O=Regiant Network Co.
Ltd/OU=Research Apartment 5/CN=CA test server
-----BEGIN CERTIFICATE-----
MIIEsTCCBFugAwIBAgIKEffFFwABAAAAPjANBgkqhkiG9w0BAQUFADCBrDEhMB8G
CSqGSIb3DQEJARYSZGluZ2pzQHN0YXItbmV0LmNuMQswCQYDVQQGEwJDTjEPMA0G
A1UECBMGRnVKaWFuMQ8wDQYDVQQHEwZGdVpob3UxIDAeBgNVBAoTF1JlZ2lhbnQg
TmV0d29yayBDby4gTHRkMR0wGwYDVQQLExRSZXNlYXJjaCBBcGFydG1lbnQgNTEX
MBUGA1UEAxMOQ0EgdGVzdCBzZXJ2ZXIwHhcNMDUwNDEyMDkyOTUzWhcNMDYwNDEy
MDkzOTUzWjCBpDEhMB8GCSqGSIb3DQEJARYSZGluZ2pzQHN0YXItbmV0LmNuMQsw
CQYDVQQGEwJDTjEPMA0GA1UECBMGRnVKaWFuMQ8wDQYDVQQHEwZGdVpob3UxIDAe
BgNVBAoTF1JlZ2lhbnQgTmV0d29yayBDby4gTHRkMR0wGwYDVQQLExRSZXNlYXJj
aCBBcGFydG1lbnQgNTEPMA0GA1UEAxMGZGluZ2pzMFwwDQYJKoZIhvcNAQEBBQAD
SwAwSAJBAM0sOymB/5v35vnf/PlJX+aqZpH9drtevsNaHkj4i3XdaJ55rFo2wLT0
qpWTI0nu638ktUa4dEIfF0AQM67sP0ECAwEAAaOCAmMwggJfMA4GA1UdDwEB/wQE
AwIE8DATBgNVHSUEDDAKBggrBgEFBQgCAjAdBgNVHQ4EFgQUIWVn8+ciY7JJkOFN
7MIkcRWWpx8wgegGA1UdIwSB4DCB3YAUckOEHCsDRS2CWPWEQ3f1IapLLCGhgbKk
ga8wgawxITAfBgkqhkiG9w0BCQEWEmRpbmdqc0BzdGFyLW5ldC5jbjELMAkGA1UE
BhMCQ04xDzANBgNVBAgTBkZ1SmlhbjEPMA0GA1UEBxMGRnVaaG91MSAwHgYDVQQK
ExdSZWdpYW50IE5ldHdvcmsgQ28uIEx0ZDEdMBsGA1UECxMUUmVzZWFyY2ggQXBh
cnRtZW50IDUxFzAVBgNVBAMTDkNBIHRlc3Qgc2VydmVyghBWrgc8EKF+jEWujT8V
UjNXMH8GA1UdHwR4MHYwOKA2oDSGMmh0dHA6Ly96ai1yb3V0ZXIvQ2VydEVucm9s
bC9DQSUyMHRlc3QlMjBzZXJ2ZXIuY3JsMDqgOKA2hjRmaWxlOi8vXFx6ai1yb3V0
ZXJcQ2VydEVucm9sbFxDQSUyMHRlc3QlMjBzZXJ2ZXIuY3JsMIGsBggrBgEFBQcB
AQSBnzCBnDBLBggrBgEFBQcwAoY/aHR0cDovL3pqLXJvdXRlci9DZXJ0RW5yb2xs
L3pqLXJvdXRlcl9DQSUyMHRlc3QlMjBzZXJ2ZXIoMSkuY3J0ME0GCCsGAQUFBzAC
hkFmaWxlOi8vXFx6ai1yb3V0ZXJcQ2VydEVucm9sbFx6ai1yb3V0ZXJfQ0ElMjB0
ZXN0JTIwc2VydmVyKDEpLmNydDANBgkqhkiG9w0BAQUFAANBABSEeYlEi7fm06en
NDdkZMjYV8LZB1JjniePwy1sUEEDa2bH9ZrcpTnJ+CSkCkxXBTc5ZWZnFTiSH/Oc
uVyQ9D8=
-----END CERTIFICATE-----
Bag Attributes: <Empty Attributes>
subject=/Email=dingjs@star-net.cn/C=CN/ST=FuJian/L=FuZhou/O=Regiant Network Co.
Ltd/OU=Research Apartment 5/CN=CA test server
issuer=/Email=dingjs@star-net.cn/C=CN/ST=FuJian/L=FuZhou/O=Regiant Network Co.
Ltd/OU=Research Apartment 5/CN=CA test server
-----BEGIN CERTIFICATE-----
```

```
MIIDLjCCAtigAwIBAgIQVq4HPBChfoxFro0/FVIzVzANBgkqhkiG9w0BAQUFADCB
rDEhMB8GCSqGSIb3DQEJARYSZGluZ2pzQHN0YXItbmV0LmNuMQswCQYDVQQGEwJD
TjEPMA0GA1UECBMGRnVKaWFuMQ8wDQYDVQQHEwZGdVpob3UxIDAeBgNVBAoTF1Jl
Z2lhbnQgTmV0d29yayBDby4gTHRkMR0wGwYDVQQLExRSZXNlYXJjaCBBcGFydG1l
bnQgNTEXMBUGA1UEAxMOQ0EgdGVzdCBzZXJ2ZXIwHhcNMDUwMjI1MDg0NjAyWhcN
MDcwMzAxMDIzNjIzWjCBrDEhMB8GCSqGSIb3DQEJARYSZGluZ2pzQHN0YXItbmV0
LmNuMQswCQYDVQQGEwJDTjEPMA0GA1UECBMGRnVKaWFuMQ8wDQYDVQQHEwZGdVpo
b3UxIDAeBgNVBAoTF1JlZ2lhbnQgTmV0d29yayBDby4gTHRkMR0wGwYDVQQLExRS
ZXNlYXJjaCBBcGFydG1lbnQgNTEXMBUGA1UEAxMOQ0EgdGVzdCBzZXJ2ZXIwXDAN
BgkqhkiG9w0BAQEFAANLADBIAkEA2R8axg75UZJM3JZNREP62r5T8t31E7Y0taah
n/1XoWxvevShE8FZPQxMPo5i3nbYokzyLPjagqoX0+jMgMKVjwIDAQABo4HTMIHQ
MAsGA1UdDwQEAwIBxjAPBgNVHRMBAf8EBTADAQH/MB0GA1UdDgQWBBRyQ4QcKwNF
LYJY9YRDd/UhqkssITB/BgNVHR8EeDB2MDigNqA0hjJodHRwOi8vemotcm91dGVy
L0NlcnRFbnJvbGwvQ0ElMjB0ZXN0JTIwc2VydmVyLmNybDA6oDigNoY0ZmlsZTov
L1xcemotcm91dGVyXENlcnRFbnJvbGxcQ0ElMjB0ZXN0JTIwc2VydmVyLmNybDAQ
BgkrBgEEAYI3FQEEAwIBATANBgkqhkiG9w0BAQUFAANBAH8ufRZ2tVYO3R7YC0IF
OzmnQrjgaBN4bpmSLkxYYKtK8ZNjo0FwUL11aq6nCGp6n8Ks0dijoMxnedB2zn0a
f0w=
-----END CERTIFICATE-----
```

Caution

Caution Where, contents between “-----BEGIN CERTIFICATE-----” and “-----END CERTIFICATE-----” are certificate texts; contents between “-----BEGIN RSA PRIVATE KEY-----” and “-----END RSA PRIVATE KEY-----” are private key texts; other contents are descriptions about the PKCS12#format.

Note

the difference between CA root certificate and router certificate: The subject and issuer of CA root certificate are the same, while those of router certificate are different. Taking the above-given process as the example, the relevant information about CA root certificate is shown below:

```
subject=/Email=dingjs@star-net.cn/C=CN/ST=FuJian/L=FuZhou/O=Regiant Network Co.
Ltd/OU=Research Apartment 5/CN=CA test server
issuer=/Email=dingjs@star-net.cn/C=CN/ST=FuJian/L=FuZhou/O=Regiant Network Co.
Ltd/OU=Research Apartment 5/CN=CA test server
The subject and issuer of router certificate are shown below:
subject=/Email=dingjs@star-net.cn/C=CN/ST=FuJian/L=FuZhou/O=Regiant Network Co.
Ltd/OU=Research Apartment 5/CN=dingjs
issuer=/Email=dingjs@star-net.cn/C=CN/ST=FuJian/L=FuZhou/O=Regiant Network Co.
Ltd/OU=Research Apartment 5/CN=CA test server
```

Run the command for importing a certificate on the control console (operation instructions are given in brackets)

```
Ruijie (config)# crypto pki import pem terminal 12345678
```

```
% Enter PEM-formatted CA certificate.(prompting you to enter PEM-formatted CA root certificate
texts)
% End with a blank line or "quit" on a line by itself.(prompting you to enter blank line or
enter "quit" to exit)
(Copy the texts of CA root certificate shown in Step 1 and paste here, as shown below)
-----BEGIN CERTIFICATE-----
MIIDLjCCAtigAwIBAgIQVq4HPBChfoxFro0/FVIzVzANBgkqhkiG9w0BAQUFADCB
rDEhMB8GCSqGSIb3DQEJARYSZGluZ2pzQHN0YXItbmV0LmNuMQswCQYDVQQGEwJD
TjEPMA0GA1UECBMGRnVKaWFuMQ8wDQYDVQQHEwZGdVpob3UxIDAeBgNVBAoTF1Jl
Z2lhbnQgTmV0d29yayBDby4gTHRkMR0wGwYDVQQLExRSZXNlYXJjaCBBcGFydG1l
bnQgNTEXMBUGA1UEAxMOQ0EgdGVzdCBzZXJ2ZXIwHhcNMDUwMjI1MDg0NjAyWhcN
MDcwMzAxMDIzNjIzWjCBrDEhMB8GCSqGSIb3DQEJARYSZGluZ2pzQHN0YXItbmV0
LmNuMQswCQYDVQQGEwJDTjEPMA0GA1UECBMGRnVKaWFuMQ8wDQYDVQQHEwZGdVpo
b3UxIDAeBgNVBAoTF1JlZ2lhbnQgTmV0d29yayBDby4gTHRkMR0wGwYDVQQLExRS
ZXNlYXJjaCBBcGFydG1lbnQgNTEXMBUGA1UEAxMOQ0EgdGVzdCBzZXJ2ZXIwXDAN
BgkqhkiG9w0BAQEFAANLADBIAkEA2R8axg75UZJM3JZNREP62r5T8t31E7Y0taah
n/1XoWxvevShE8FZPQxMPo5i3nbYokzyLPjagqoX0+jMgMKVjwIDAQABo4HTMIHQ
MAsGA1UdDwQEAwIBxjAPBgNVHRMBAf8EBTADAQH/MB0GA1UdDgQWBBRyQ4QcKwNF
LYJY9YRDd/UhqkssITB/BgNVHR8EeDB2MDigNqA0hjJodHRwOi8vemotcm91dGVy
L0NlcnRFbnJvbGwvQ0ElMjB0ZXN0JTIwc2VydmVyLmNybDA6oDigNoY0ZmlsZTov
L1xcemotcm91dGVyXENlcnRFbnJvbGxcQ0ElMjB0ZXN0JTIwc2VydmVyLmNybDAQ
BgkrBgEEAYI3FQEEAwIBATANBgkqhkiG9w0BAQUFAANBAH8ufRZ2tVYO3R7YC0IF
OzmnQrjgaBN4bpmSLkxYYKtK8ZNjo0FwUL11aq6nCGp6n8Ks0dijoMxnedB2zn0a
f0w=
-----END CERTIFICATE-----
quit
Certificate has the following attributes:
Fingerprint: B286A3F4 4930D46D 81D4A544 885D611C (Fingerprint of CA root certificate)
% Do you accept this certificate? [yes/no]: yes(Prompting you to confirm the fingerprint)
% Certificate successfully imported
% Enter PEM-formatted encrypted private key.(Prompting you to enter PEM-formatted private key
texts)
% End with "quit" on a line by itself.(Prompting you to type in "quit" to exit)
(Copy the texts of private key shown in Step 1 and paste here, as shown below)
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4,ENCRYPTED
DEK-Info: DES-EDE3-CBC,251F9D955610C376
GDG2s1mbs/MJCpo5w2bu972jK1OZYtv3RQunH4I29c9H5uq3LtyvNA9RwrlpRQ3t
iUmkvQrU3/6SBp4Rqx1EU2UWgv1KRqqYwRVbdPdBZYVJLrso3Ov/9eaS4TiD+4Dl
NfJ1sAA4OONdVKDCLcGZIB43Wq5rAlqzsyjcF6tx3fWsSankVjQfroTv7UvP+ijj
uGndmJwbXEiATxlt+Smvt2/CGjr8nIC55T1W+tW0itkBdZhnvBJekOFM4BdgoLZc
3vueTIHmTurHvvdLIytYjQHsxVsf3vRGMcQhohM98nAYsIDBil4OIh1hc+ZnhGsn
TFLPMmMuJnBWMYopfaMPNrcdbpu+n4Qj2QiRoVTEoI7P1IAY/Oa2uc+kDuUX3KlW
sQQPnFNiU0Q/T9BrsolxI2Wkak7cvaNxbmhuU+5wNUGybQfcfP3CWg==
-----END RSA PRIVATE KEY-----
```

```
quit
% RSA private key successfully imported
Enter the base 64 encoded certificate.(Prompting you to enter PEM-formatted router certificate
texts)
End with a blank line or the word "quit" on a line by itself(prompting you to enter blank line
or enter "quit" to exit)
```

(Copy the texts of router certificate shown in Step 1 and paste here, as shown below)

```
-----BEGIN CERTIFICATE-----
MIIEsTCCBFugAwIBAgIKEffFFwABAAAAPjANBgkqhkiG9w0BAQUFADCBrDEhMB8G
CSqGSIb3DQEJARYSZGluZ2pzQHN0YXItbmV0LmNuMQswCQYDVQQGEwJDTjEPMA0G
A1UECBMGRnVKaWFuMQ8wDQYDVQQHEwZGdVpob3UxIDAeBgNVBAoTF1JlZ2lhbnQg
TmV0d29yayBDby4gTHRkMR0wGwYDVQQLExRSZXNlYXJjaCBBcGFydG1lbnQgNTEX
MBUGA1UEAxMOQ0EgdGVzdCBzZXJ2ZXIwHhcNMDUwNDEyMDkyOTUzWhcNMDYwNDEy
MDkzOTUzWjCBpDEhMB8GCSqGSIb3DQEJARYSZGluZ2pzQHN0YXItbmV0LmNuMQsw
CQYDVQQGEwJDTjEPMA0GA1UECBMGRnVKaWFuMQ8wDQYDVQQHEwZGdVpob3UxIDAe
BgNVBAoTF1JlZ2lhbnQgTmV0d29yayBDby4gTHRkMR0wGwYDVQQLExRSZXNlYXJj
aCBBcGFydG1lbnQgNTEPMA0GA1UEAxMGZGluZ2pzMFwwDQYJKoZIhvcNAQEBBQAD
SwAwSAJBAM0sOymB/5v35vnf/PlJX+aqZpH9drtevsNaHkj4i3XdaJ55rFo2wLT0
qpWTI0nu638ktUa4dEIfF0AQM67sP0ECAwEAAaOCAmMwggJfMA4GA1UdDwEB/wQE
AwIE8DATBgNVHSUEDDAKBggrBgEFBQgCAjAdBgNVHQ4EFgQUIWVn8+ciY7JJkOFN
7MIkcRWWpx8wgegGA1UdIwSB4DCB3YAUckOEHCsDRS2CWPWEQ3f1IapLLCGhgbKk
ga8wgawxITAfBgkqhkiG9w0BCQEWEmRpbmdqc0BzdGFyLW5ldC5jbjELMAkGA1UE
BhMCQ04xDzANBgNVBAgTBkZ1SmlhbjEPMA0GA1UEBxMGRnVaaG91MSAwHgYDVQQK
ExdSZWdpYW50IE5ldHdvcmsgQ28uIEx0ZDEdMBsGA1UECxMUUmVzZWFyY2ggQXBh
cnRtZW50IDUxFzAVBgNVBAMTDkNBIHRlc3Qgc2VydmVyghBWrgc8EKF+jEWujT8V
UjNXMH8GA1UdHwR4MHYwOKA2oDSGMmh0dHA6Ly96ai1yb3V0ZXIvQ2VydEVucm9s
bC9DQSUyMHRlc3QlMjBzZXJ2ZXIuY3JsMDqgOKA2hjRmaWxlOi8vXFx6ai1yb3V0
ZXJcQ2VydEVucm9sbFxDQSUyMHRlc3QlMjBzZXJ2ZXIuY3JsMIGsBggrBgEFBQcB
AQSBnzCBnDBLBggrBgEFBQcwAoY/aHR0cDovL3pqLXJvdXRlci9DZXJ0RW5yb2xs
L3pqLXJvdXRlcl9DQSUyMHRlc3QlMjBzZXJ2ZXIoMSkuY3J0ME0GCCsGAQUFBzAC
hkFmaWxlOi8vXFx6ai1yb3V0ZXJcQ2VydEVucm9sbFx6ai1yb3V0ZXJfQ0ElMjB0
ZXN0JTIwc2VydmVyKDEpLmNydDANBgkqhkiG9w0BAQUFAANBABSEeYlEi7fm06en
NDdkZMjYV8LZB1JjniePwy1sUEEDa2bH9ZrcpTnJ+CSkCkxXBTc5ZWZnFTiSH/Oc
uVyQ9D8=
-----END CERTIFICATE-----
quit
% Certificate successfully imported
Ruijie (config)#
```

Caution

Caution Currently, Ruijie products can only support one CA root certificate and one router certificate. If another certificate is imported, the previous one is overwritten.

The import is only considered successful when CA root certificate, private key and router certificate are all successfully imported, and device configurations will be updated then, or else all import operations will be cancelled without causing any impact on the configurations. For example, when the import of router certificate fails, the import of CA root certificate and private key will all be cancelled

While copying and pasting certificate texts, do not omit any character, especially the "begin" and "end" lines. After successful import, execute the "write" operation to store the certificate and private key into FLASH, or else the import contents will be lost after shutdown.

The fingerprint of CA root certificate is used to avoid the artificial falsification of CA root certificate during the process of transmission. The CA will issue fingerprint information (including fingerprint and the hmac algorithm for fingerprint calculation) together with the CA root certificate. The user of CA root certificate must use the same fingerprint algorithm used by CA to calculate fingerprint, and check with the fingerprint issued by the CA. RGOS supports the fingerprint algorithm of SHA-1.

Verify the result of certificate import. After a successful import, the certificate will be stored in DER-encoded hexadecimal format in the system configuration file. Run the **show running** command to query the serial number and DER encoded contents of the certificate (refer to the "Configuration Example" section), or you can also run the **show crypto pki cert** command (see the "Monitoring and Maintenance" section). Note that the private key will not be displayed.

Acquiring Router Certificate Through SCEP

To configure a trustpoint, run the following commands:

Command	Function
Ruijie (config)#**crypto pki trustpoint** *CA_name*	Enters trustpoint configuration mode. *CA_name* is the common name of the CA corresponding to this trustpoint, namely the character string entered in the **Common name for this CA** field in Step 9 of the "Installing the Certificate Services on a Windows 2003 Server" section.
Ruijie (ca-trustpoint)# **asymmetric** sm2	Specifies a certificate algorithm and uses RSA (optional) in a default condition.
Ruijie (ca-trustpoint)#**enrollment url** *http://192.168.50.203/certsrv/mscep/mscep.dll*	Configures the certificate enrollment URL of this trustpoint, namely the URL shown in step 8 of the "Installing SCEP add-on" section (domain name); if there is no DNS server, an IP address can also be used; manually replace the domain name with the corresponding IP address.
Ruijie (ca-trustpoint)#**enrollment retry period** *number*	Configures the polling period when this trustpoint enters the polling state during certificate enrollment. Number is a numeric value (unit: minute); the default value is one minute.
Ruijie (ca-trustpoint)#**enrollment retry count** *number*	Configures the polling count when this trustpoint enters the polling state during certificate enrollment. Number is a numeric value (unit: time); the default value is 60 times.
Ruijie (ca-trustpoint)# **enrollment auto-enroll** *percentage*	Specifies the update period of the certificate corresponding to the trustpoint; percentage ranges from 1 through 100 to specify when the certificate will be updated.
Ruijie (ca-trustpoint)# **enrollment renewable**	Enables the CA server corresponding to the trustpoint to support certificate update

Command	Function
Ruijie (ca-trustpoint)#**exit**	Exits trustpoint configuration mode.

To acquire a CA root certificate, run the following command:

Command	Function
Ruijie (config)# **crypto pki authenticate** *CA_name*	Acquires the CA root certificate.

To register a certificate, run the following command:

Command	Function
Ruijie (config)# **crypto pki enroll** *CA_name*	Enrolls trustpoint and acquires the router certificate corresponding to this trustpoint.

The process of acquiring a router certificate through SCEP is as follows:

Ensure the time matches with standard time on each device. Use the **show clock** command on the router to ensure that the time is correct.

Configure an IP address to ensure that the router can access the CA server.

Configure the hostname of the router

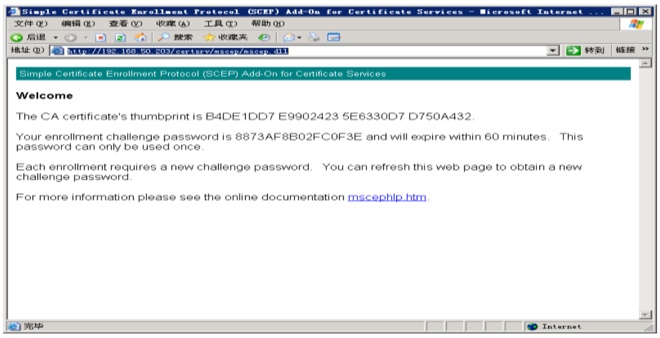
```
ruijie(config)#hostname router
router(config)#
Generate a key
router(config)#crypto pki key generate rsa
gernerate-key
Choose the size of the key modulus in the range of 360 to 2048 for your General Purpose Keys.
Choosing a key modulus greater than 512 may take a few minutes.
How many bits in the modulus:1024  //Modulus size of the key
Configure the trustpoint corresponding to the CA
router(config)#crypto pki trustpoint CA
router(ca-trustpoint)#enrollment url http://192.168.50.203/certsrv/mscep/mscep.dll //URL
for enrollment
router(ca-trustpoint)#exit
```

Acquire and verify the CA root certificate.

```
router(config)#crypto pki authenticate CA
Certificate has the following attributes:
MD5 fingerprint: B4DE1DD7 E9902423 5E6330D7 D750A432
SHA1 fingerprint: AD070162 672A7C57 BD5EE522 A95AAFA1 351524D0
% Do you accept this certificate?[yes/no]:yes   //Enter yes to accept the CA certificate
Trustpoint CA certificate accepted.
```

To acquire the fingerprint and passphrase of the certificate, visit: http://ca-ip-address/certsrv/mscep/mscep.dll.

The following webpage will be displayed, prompting you to proceed with authentication. Make sure that the administrator can visit this website using the fingerprint and passphrase.


Simple Certificate Enrollment Protocol (SCEP) Add-On for Certificate Services - Microsoft Internet ...
文件(F) 编辑(E) 查看(V) 收藏(A) 工具(T) 帮助(H)
后退 搜索 收藏夹
地址(D) http://192.168.50.203/certsrv/mscep/mscep.dll
转到 链接
完毕
Internet

Simple Certificate Enrollment Protocol (SCEP) Add-On for Certificate Services

Welcome

The CA certificate's thumbprint is B4DE1DD7 E9902423 5E6330D7 D750A432.

Your enrollment challenge password is 8873AF8B02FC0F3E and will expire within 60 minutes. This password can only be used once.

Each enrollment requires a new challenge password. You can refresh this web page to obtain a new challenge password.

For more information please see the online documentation mscephlp.htm.

Now the fingerprint and passphrase are acquired. You need to enter the following contents on the router (replies from CA are also provided for your reference)

router(config)#crypto pki enroll CA

```
%
%Start certificate enrollment ..
%Create a challenge password. You will need to verbally provide this
   password to the CA Administrator in order to revoke your certificate.
   For security reasons your password will not be saved in the configuration.
   Please make a note of it.

Password:F4EEE4FEB3766007    //Enter the password acquired from CA
Re-enter password:F4EEE4FEB3766007
%The subject name in the certificate will include: router
```

Display the state of the trustpoint

router(config)#show crypto pki trustpoints CA status

```
Trustpoint CA Status:
  Issuing CA certificate configured:
    Subject Name:
      /CN=CA
    Fingerprint MD5: B4DE1DD7 E9902423 5E6330D7 D750A432
    Fingerprint SHA1: AD070162 672A7C57 BD5EE522 A95AAFA1 351524D0
  Router General Purpose certificate pending:    Requested Subject Name:
      /unstructuredName=router
    Request Fingerprint MD5: E8F50E2A 1FB46B00 52C6A9DB CC20AA42
Request Fingerprint SHA1: B1031204 E7A2D547 CC48F0A6 2BAE8420 829637D8
    Enrollment polling: 2 times (58 left)
  Last enrollment status: Pending    //In pending state
  State:
    Keys generated ............. Generated
    Issuing CA authenticated ....... Yes
Certificate request(s) ..... Pending
```

When "Certificate requests pending" shows up (as shown above), check "Pending Requests" on the CA server and issue this certificate. Generally, the request will take 10 to 15 seconds.

Right-click the certificate and choose **All Tasks** > **Issue**.

The certificate will then be moved to the "Issued Certificate" node. Return to the router console and query trustpoint status.

router(config)#show crypto pki trustpoints CA status

```
Trustpoint CA Status:
  Issuing CA certificate configured:
    Subject Name:
      /CN=CA
    Fingerprint MD5: B4DE1DD7E 99024235 E6330D7D 750A432
    Fingerprint SHA1: AD070162 672A7C57 BD5EE522 A95AAFA1 351524D0
  Router General Purpose certificate configured:
    Subject Name:
      /unstructuredName=router
    Fingerprint MD5: 7FFF7F4C 225850A3 D0D39EA3 EFAD8D5A
    Fingerprint SHA1: 84E3678C F2DE94DD 63397145 87CDC9C6 1010A82F
  Last enrollment status: Granted    //Certificate application is successful
  State:
    Keys generated ............. Generated
    Issuing CA authenticated ....... Yes
    Certificate request(s) ..... Yes
```

Configuring SNC certificate

Configure trustpoint:

Command	Function
Ruijie (config)#**crypto pki trustpoint** *CA_name*	Enters trustpoint configuration mode. *CA-name* is the common name of the corresponding CA, namely, the string entered in the **CA common name (C)** field in step 9 in the “Installing Certificate Services on a Windows 2003 Server” section.
Ruijie (ca-trustpoint)#**enrollment url** *http://192.168.50.203/certsrv/mscep/mscep.dll* **auto-up**	Configures the certificate enrollment URL of this trustpoint, namely the URL shown in step 8 of the "Installing SCEP add-on" section; if there is no DNS server, an IP address can also be used; manually replace the domain name with the corresponding IP address. adds **auto-up** in this command to get automatically generated certificate.
Ruijie (ca-trustpoint)#**exit**	Exits trustpoint configuration mode.

The configuration of getting SNC certificate and SECP certificate are similar. The only difference lies in **auto-up** at the end of SNC certificate enrollment URL configuration., as shown below:

```
router(config)#crypto pki trustpoint CA
router(ca-trustpoint)#enrollment url http://192.168.50.203/certsrv/mscep/mscep.dll auto-up
//enrollment url
router(ca-trustpoint)#exit
```

Configuring an Offline Certificate

Configure trustpoint:

Command	Function
Ruijie (config)#**crypto pki trustpoint** *CA_name*	Enters trustpoint configuration mode. *CA-name* is the common name of the corresponding CA, namely, the string entered in the **CA common name (C)** field in step 9 in the "Installing Certificate Services on a Windows 2003 Server" section.
Ruijie (ca-trustpoint)# **asymmetric** sm2	Specifies a certificate algorithm and uses RSA (optional) in a default condition.
Ruijie (ca-trustpoint)# **enrollment offline subject**	Configures the unique name of the router.
Ruijie (ca-trustpoint)#**exit**	Exits trustpoint configuration mode.

To register a certificate, run the following command:

Command	Function
Ruijie (config)# **crypto pki enroll** *CA_name*	Registers trustpoint and obtains the router certificate corresponding to the trustpoint.

The process for configuring an offline certificate is as follows:

- Step 1 Generate an RSA public/private key pair (mandatory).
- Step 2 Define a CA (mandatory).
- Step 3 Register an offline certificate (mandatory).
- Step 4 The CA issues a certificate (mandatory).
- Step 5 Import the certificate (mandatory).

Generate an RSA key pair:

```
Ruijie (config)#crypto pki key generate rsa //Generate a key.
Ruijie (config)#crypto pki trustpoint CA_name //Define a CA. CA-name: FQDN of the CA, which will be provided by the CA administrator
ruijie(ca-trustpoint)#enrollment offline subject //Set DN information of the offline certificate
```

You are about to be asked to enter you Distinguished Name(DN) information that will be incorporated into your certificate request. There are quite a few fields but you can leave some blank.

Common Name (eg, YOUR name) []: //Your first name and last name

Organizational Unit Name (eg, section) []: //Your organizational unit name

Organization Name (eg, company) []: //Your organization name

Locality Name (eg, city) []: //Your city or region

State or Province Name (full name) []: //Your state or province

Country Name (2 letter code) [CN]: //2-letter country code of your organization

```
The subject name is: cn=fhsjflsdgingsd,ou=research,o=ruijie,l=CD,st=SC,c=CN
Is it correct[yes/no]:yes
ruijie(config)#crypto pki enroll CA_name          //Register an offline certificate
%The subject name in the certificate will include:
cn=fhsjflsdgingsd,ou=research,o=ruijie,l=CD,st=SC,c=CN
-----BEGIN CERTIFICATE REQUEST-----
MIIBpDCCAQ0CAQAwZDEXMBUGA1UEAxMOZmhzamZsc2RnaW5nc2QxETAPBgNVBAsT
CHJlc2VhcmNoMQ8wDQYDVQQKEwZydWlqaWUxCzAJBgNVBAcTAkNEMQswCQYDVQQI
EwJTQzELMAkGA1UEBhMCQ04wgZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBANep
/WFrf0uBBp7ZIPMC7Dq22mUtzc3xWrT3V5sk/P98+KTXlKYy7aYCKZqhgCw/5XHP
6fAV9d7kKcs9ynptjbagjdFpeWSRpRzJ0U+fYg1muJf7U3ZuyFMBOQgwOvofwcOa
sJ53RhmadZqdAHzPDtQT9XVbl4tSNYckGiOm3My3AgMBAAGgADANBgkqhkiG9w0B
AQQFAAOBgQBFAC/oAVXrKpVvks0Mvk+84bKIR0tY2opqyRo9Ax26rZM8hK4oULQS
n3Ar7O3pBoWt1YbX0ZpUpEgu1IRCm0PwIeQ6uN6KwnO3a6A3AMLgWrwQ29rn7kQG
JbsHZ+Okk80CzZu6s8OBtasB6VU4LFCGwBAtbL83Syp973c8cYGPWg==
-----END CERTIFICATE REQUEST-----
Copy the contents (PKCS10 format) between “-----BEGIN CERTIFICATE REQUEST-----“ and “-----END
CERTIFICATE REQUEST-----” to the CA to issue a certificate.
% Enter PEM-formatted CA certificate.
% End with a blank line or "quit" on a line by itself.
```

//Paste the CA root certificate in PEM format.

Certificate has the following attributes:

```
MD5 fingerprint: D869FAEE D797E625 B248217D 2050BF48
SHA1 fingerprint: D0B10C45 751402F0 646B4DBF E5B26AE2 74207498
%% Do you accept this certificate?[yes/no]:yes
% CA Certificate successfully imported
% Enter PEM-formatted certificate.
% End with a blank line or "quit" on a line by itself.
```

//Paste the router certificate in PEM format issued by the CA.

```
% Router Certificate successfully imported
```

Certificate Configuration Commands (Optional)

Certificate configuration commands include certificate chain configuration commands and certificate configuration commands.

To create a certificate chain, run the following command in global configuration mode. Use the **no** form of this command to delete a certificate chain:

Command	Function

Ruijie(config)# **crypto pki certificate chain**	Creates a router certificate chain and enters certificate chain configuration mode (config_cert_chain).

To add a certificate to the certificate chain, run the following command in certificate chain configuration mode. Use the **no** form of this command to delete the certificate:

Command	**Function**
Ruijie(config_cert_chain)# **certificate** [**CA**] *serial_num*	Enters certificate configuration mode (config-pubkey) in order to enter certificate data with the specified serial number; *serial_num* is the serial number of the certificate; The *CA* parameter indicates the CA root certificate.

To enter certificate data, run the following command line by line in certificate configuration mode (config-pubkey).

Command	**Function**
Ruijie(config-pubkey)# *308202E6 30820290 A0030201 0202107F FFBB3997 39B4814B E16B4FF9 067A4B30*	Enters certificate data.

To exit certificate configuration mode, type in "quit" and the system will immediately analyze and verify the certificate data. If the certificate data is illegal, the contents entered will be cancelled. Note that "exit" and "Ctrl+Z" do not take effect in certificate configuration mode.

Command	**Function**
Ruijie(config-pubkey)# **quit**	Ends certificate data input and exits certificate configuration mode.

Caution

The commands configured in this section are only used to store, display and delete certificates. It is not recommended that you configure the certificate by entering certificate data line by line manually, as it is quite troublesome. Refer to relevant instructions for certificate import. After a successful import, run the **show running** command and you will see that the system has automatically created the certificate chain and converted CA certificate and router certificate into the format shown herein as system configurations. Actually, as Ruijie products do not provide the command for separately configuring private keys, you can only import the certificate as per relevant instructions.

If you want to manually configure the certificate by using the **certificate** command, configure the CA root certificate first and then configure the router certificate, as the former one will be needed for verification while configuring the later one.

When using the **no** form of the command to delete a CA root certificate or certificate chain, the CA root certificate, router certificate and private key in system configuration all will be deleted.

To check certificate configurations, run the **show running** command or the **show crypto pki cert** command. For details, refer to the "Monitoring and Maintenance" section.

Example of manually configuring a CA root certificate:

```
Ruijie(config)# crypto pki certificate chain
Ruijie(config-cert-chain)# certificate ca
7FFFBB399739B4814BE16B4FF9067A4B
Ruijie(config-pubkey)# 308202E6 30820290 A0030201 0202107F FFBB3997 39B4814B E16B4FF9
067A4B30
Ruijie(config-pubkey)# 0D06092A 864886F7 0D010105 05003081 8F312330 2106092A 864886F7
0D010901
Ruijie(config-pubkey)# 1614776C 6370796A 77624073 7461722D 6E65742E 636E310B 30090603
55040613
Ruijie(config-pubkey)# 02434E31 0B300906 03550408 1302666A 310F300D 06035504 07130666
757A686F
Ruijie(config-pubkey)# 75311230 10060355 040A1309 52656420 4769616E 74311530 13060355
040B130C
Ruijie(config-pubkey)# 44657061 72746D65 6E742035 31123010 06035504 03130943 41205365
72766572
Ruijie(config-pubkey)# 301E170D 30353036 32323035 34363332 5A170D30 37303632 32303535
3434355A
Ruijie(config-pubkey)# 30818F31 23302106 092A8648 86F70D01 09011614 776C6370 796A7762
40737461
Ruijie(config-pubkey)# 722D6E65 742E636E 310B3009 06035504 06130243 4E310B30 09060355
04081302
Ruijie(config-pubkey)# 666A310F 300D0603 55040713 0666757A 686F7531 12301006 0355040A
13095265
Ruijie(config-pubkey)# 64204769 616E7431 15301306 0355040B 130C4465 70617274 6D656E74
20353112
Ruijie(config-pubkey)# 30100603 55040313 09434120 53657276 6572305C 300D0609 2A864886
F70D0101
Ruijie(config-pubkey)# 01050003 4B003048 024100BE D1E81427 7A302B5E 11CA43FD 2F2B7EA9
8A0796A2
Ruijie(config-pubkey)# CFFE9DB7 D3DA54C3 034AA844 B3F011DC 8ABB7253 9758B13F DF6B8A9E
5F46D300
Ruijie(config-pubkey)# 402E24D3 85A74142 55F77502 03010001 A381C530 81C2300B 0603551D
0F040403
Ruijie(config-pubkey)# 0201C630 0F060355 1D130101 FF040530 030101FF 301D0603 551D0E04
16041464
Ruijie(config-pubkey)# 4612C027 A49E010C 65DAF86E E7FEC656 ECADD430 71060355 1D1F046A
30683031
Ruijie(config-pubkey)# A02FA02D 862B6874 74703A2F 2F7A6A2D 726F7574 65722F43 65727445
6E726F6C
Ruijie(config-pubkey)# 6C2F4341 25323053 65727665 722E6372 6C3033A0 31A02F86 2D66696C
653A2F2F
Ruijie(config-pubkey)# 5C5C7A6A 2D726F75 7465725C 43657274 456E726F 6C6C5C43 41253230
53657276
Ruijie(config-pubkey)# 65722E63 726C3010 06092B06 01040182 37150104 03020100 300D0609
2A864886
```

```
Ruijie(config-pubkey)# F70D0101 05050003 4100342F 8D936843 607B685F F07E910C 5CE35898
7C5395AE
Ruijie(config-pubkey)# C2B81CFF 82A4AE95 A881A88A FFF96F92 723EFA6F 847D8347 930F8576
48AE68B9
Ruijie(config-pubkey)# 5A72CF09 50BE1BA7 E187
Ruijie(config-pubkey)# quit
```

Configuring Certificate Revocation Check Policy (Optional)

When checking whether the certificate of the communication peer is valid, Ruijie products provide strict and loose verification. In strict verification mode, the certificate must be verified for revocation. If the correct CRL is not found, the peer certificate will not be accepted; in loose verification mode, the certificate will not be verified for revocation. By default, the strict mode will be used. Run the following command in global configuration mode, you can change the check policy to loose mode, and use the **no** form of this command to restore to strict mode.

Command	Function
Ruijie(config)# **crypto pki revocation-check none**	When this command is used, there is no need to check whether the certificate has been revoked according to CRL while checking the certificate of communication peer.

Note

The check policy shall be determined according to the fact that whether the device may receive the revoked peer certificate. For example, when the certificate is used in IKE center-branch network model, as the central device needs to accept the negotiation requests initiated by many dialers, and some certificates may have been revoked, the strict mode must be configured then to avoid unauthorized access. The branch devices only initiate the negotiation attempt with the central device, and are not possible to receive a revoked certificate. Therefore, the loose mode will be sufficient and network resources needed for CRL update can also be saved.

Downloading a CRL (Optional)

By default, strict certificate revocation is used. At this time, you must download a CRL. The maximum size of a CRL file allowed by the RGOS is 1 MB; otherwise, CRL download will be rejected. On Ruijie products, a CRL file can be downloaded through HTTP from a URL obtained in the following methods (priority arranged in descending order):

15) Specified by using the **crypto pki crl url** *http://www.myca.cn/certsrv/certcrl.crl* command
16) Extension of CRL distribution point of CA root certificate configured on the device;
17) Extension of CRL distribution point of router certificate configured on the device;

CRL can be downloaded by the following means:

- Manually download CRL by using the **crypto pki crl request** command;
- When CA root certificate and router certificate are configured and strict mode is adopted for certificate check, the CRL will be detected every one minute for presence and expiration, and will be downloaded automatically;
- When strict mode is adopted for certificate check, the system will verify whether the local CRL has expired or not and download immediately during certificate check if the local CRL has expired.

Note

Note When a digital certificate is no longer needed by the device, delete relevant configurations or configure a loose certificate revocation check policy for the following considerations:

1. CRL expiration check executed once every minute can be saved;
2. If the CRL file is large, automatic update will consume certain network resources and occupy FLASH space and memory space.

To manually specify the URL for downloading CRL, run the following command in global configuration mode; use the **no** form of this command to delete this configuration:

Command	Function
Ruijie(config)# **crypto pki crl url** *url_string*	Manually specifies the URL for downloading a CRL file.

Caution

url_string must begin with http://; port 80 is used as the downloading port by default, or else you must specify a port after the domain name, for example, http://www.myca.cn:1020/; the directory name is **certsrv** by default, or you can specify a directory by running the **http://www.myca.cn***/CertDir/* command; the CRL file is **certcrl.crl** by default, or you can specify it by running the **http://www.myca.cn/certsrv/***mycertcrl.crl*; the value of *url_string* must contain no space; if your URL must contain spaces, you can type in "%20" instead, for example, http://www.myca.cn/certsrv/CA%20Server%20Crl.crl.

The domain of url_string can use an IP address directly, such as http://202.101.211.123, or an internal host name, such as http://myserver. No matter the URL is obtained through manual configuration or certificate, the device will automatically proceed with domain name resolution or host name resolution while starting to download CRL file. Make sure the relevant configurations are correct. If domain name resolution is needed, the correct DNS server address must be configured; if internal host name resolution is needed, use the **ip host** command to configure the IP address of the host.

The extension of CRL distribution point may contain multiple URLs. RGOS can only use one URL. Pay attention to this issue during CA server configuration.

To manually download a CRL, run the following command in global configuration mode:

Command	Function
Ruijie(config)# **crypto pki crl request**	Manually starts CRL download; start CRL download according to the currently configured certificate and URL (this command cannot be stored).

During CRL download, run the **crypto pki crl request** command and the system will prompt that the download process has started. Upon successful download, the message “%Crl download and decode successfully!” will be displayed on the console; use the **dir** command to check the file in FLASH and its creation time, or you can check the result of CRL download, as shown below:

```
Ruijie# dir
Directory of flash:/
```

```
5    an      68 0xdbc28957 Jan  1 2005 00:00:00 tftp_config.bin
8    an  4301816 0x3e415b47 Jun 28 2005 15:03:46 rgos.bin
20    an     5311 0xeaa56cb0 Jul  4 2005 10:04:37 config.text
26    an      427 0x5bd43f32 Jun 29 2005 10:00:07 certcrl.crl
Ruijie# show clock
clock: 2005-6-29 10:0:19
```

The name of the CRL file is **certcrl.crl**; the download time is 2005-6-29 10:00:07; the current time is 10:00:19. We can see that this CRL was downloaded just now.

In addition, Ruijie products also allow you to use the **show crypto pki crl** command to query information about the present CRL file (see the "Monitoring and Maintenance" section).

Configuration Example

This section shows the outputs of the **show running** command after completing certificate configuration. It shall be noted that the private key will not be displayed in the system configuration file as it is considered private information. Therefore, certificate configuration cannot be completed by copying and pasting the following configurations to the console or running the **copy tft flash** command to copy the configuration file to **config.txt**. To configure a certificate, you must run the **crypto pki import pem terminal** Command to import the certificate. This example only shows the visible configuration results.

Caution

For how the certificate application module uses the digital certificate, refer to instructions on relevant application modules.

Configuration example is shown below:

```
Ruijie# sh run
Building configuration...
Current configuration : 5331 bytes
!
version 8.31(building 1)
hostname Ruijie
!
crypto pki certificate chain
certificate ca 7FFFBB399739B4814BE16B4FF9067A4B
308202E6 30820290 A0030201 0202107F FFBB3997 39B4814B E16B4FF9 067A4B30
0D06092A 864886F7 0D010105 05003081 8F312330 2106092A 864886F7 0D010901
1614776C 6370796A 77624073 7461722D 6E65742E 636E310B 30090603 55040613
02434E31 0B300906 03550408 1302666A 310F300D 06035504 07130666 757A686F
75311230 10060355 040A1309 52656420 4769616E 74311530 13060355 040B130C
44657061 72746D65 6E742035 31123010 06035504 03130943 41205365 72766572
301E170D 30353036 32323035 34363332 5A170D30 37303632 32303535 3434355A
30818F31 23302106 092A8648 86F70D01 09011614 776C6370 796A7762 40737461
722D6E65 742E636E 310B3009 06035504 06130243 4E310B30 09060355 04081302
666A310F 300D0603 55040713 0666757A 686F7531 12301006 0355040A 13095265
```

```
64204769 616E7431 15301306 0355040B 130C4465 70617274 6D656E74 20353112
30100603 55040313 09434120 53657276 6572305C 300D0609 2A864886 F70D0101
01050003 4B003048 024100BE D1E81427 7A302B5E 11CA43FD 2F2B7EA9 8A0796A2
CFFE9DB7 D3DA54C3 034AA844 B3F011DC 8ABB7253 9758B13F DF6B8A9E 5F46D300
402E24D3 85A74142 55F77502 03010001 A381C530 81C2300B 0603551D 0F040403
0201C630 0F060355 1D130101 FF040530 030101FF 301D0603 551D0E04 16041464
4612C027 A49E010C 65DAF86E E7FEC656 ECADD430 71060355 1D1F046A 30683031
A02FA02D 862B6874 74703A2F 2F7A6A2D 726F7574 65722F43 65727445 6E726F6C
6C2F4341 25323053 65727665 722E6372 6C3033A0 31A02F86 2D66696C 653A2F2F
5C5C7A6A 2D726F75 7465725C 43657274 456E726F 6C6C5C43 41253230 53657276
65722E63 726C3010 06092B06 01040182 37150104 03020100 300D0609 2A864886
F70D0101 05050003 4100342F 8D936843 607B685F F07E910C 5CE35898 7C5395AE
C2B81CFF 82A4AE95 A881A88A FFF96F92 723EFA6F 847D8347 930F8576 48AE68B9
5A72CF09 50BE1BA7 E187
quit
!
certificate 162A7A1D000000000002
308204F9 308204A3 A0030201 02020A16 2A7A1D00 00000000 02300D06 092A8648
86F70D01 01050500 30818F31 23302106 092A8648 86F70D01 09011614 776C6370
796A7762 40737461 722D6E65 742E636E 310B3009 06035504 06130243 4E310B30
09060355 04081302 666A310F 300D0603 55040713 0666757A 686F7531 12301006
0355040A 13095265 64204769 616E7431 15301306 0355040B 130C4465 70617274
6D656E74 20353112 30100603 55040313 09434120 53657276 6572301E 170D3035
30363232 30353530 34385A17 0D303630 36323230 36303034 385A3081 80311630
1406092A 864886F7 0D010901 16077A68 616F6A75 6E310B30 09060355 04061302
434E310B 30090603 55040813 02666A31 0F300D06 03550407 13066675 7A686F75
31123010 06035504 0A130952 65642047 69616E74 31153013 06035504 0B130C44
65706172 746D656E 74203531 10300E06 03550403 13077A68 616F6A75 6E308201
22300D06 092A8648 86F70D01 01010500 0382010F 00308201 0A028201 0100C6E2
7AE88D6D D8BB56A8 9C036214 E52E23E5 A526313D B22465B1 F2CC07E3 EFCC023C
D06E008D FCCE3AB6 457ACBA0 87941FC3 9243366A B27C9CD5 CA7E83BA 76497FBE
F41F4AA1 0B982296 E27954A0 ED1C6230 B7EE6A6E CB72E99C D9E8B0DC F5C6198F
2B2A85FA BFFF0840 7EF2A1DF D18BEF68 321E1A45 FA16DE33 B06290BD 9C8EEC7C
6E494875 E65CCEB1 8E1C80F3 5B796CA1 31B2A948 379FED45 9585BA98 0F42C578
4C3DA245 73903D0B 1A7C53B5 971AA643 2F44540F A1513A0E 9F8B2ED1 70CB3699
9157D2B7 9D7CEE07 CF4AC7CD 71DCCE72 DC75A003 B236BE8E AFCA9946 038327D3
FF241E4C 0C2199B4 FE5A4D61 B5E9B438 DC592C37 F39302FC 0988021B D0450203
010001A3 82022430 82022030 0E060355 1D0F0101 FF040403 0206C030 13060355
1D25040C 300A0608 2B060105 05080202 301D0603 551D0E04 16041484 7E33A391
A5261D2D BB5465BF C72A2A2E 87D5A930 81CB0603 551D2304 81C33081 C0801464
4612C027 A49E010C 65DAF86E E7FEC656 ECADD4A1 8195A481 9230818F 31233021
06092A86 4886F70D 01090116 14776C63 70796A77 62407374 61722D6E 65742E63
6E310B30 09060355 04061302 434E310B 30090603 55040813 02666A31 0F300D06
03550407 13066675 7A686F75 31123010 06035504 0A130952 65642047 69616E74
31153013 06035504 0B130C44 65706172 746D656E 74203531 12301006 03550403
```

```
13094341 20536572 76657282 107FFFBB 399739B4 814BE16B 4FF9067A 4B307106
03551D1F 046A3068 3031A02F A02D862B 68747470 3A2F2F7A 6A2D726F 75746572
2F436572 74456E72 6F6C6C2F 43412532 30536572 7665722E 63726C30 33A031A0
2F862D66 696C653A 2F2F5C5C 7A6A2D72 6F757465 725C4365 7274456E 726F6C6C
5C434125 32305365 72766572 2E63726C 30819806 082B0601 05050701 0104818B
30818830 4106082B 06010505 07300286 35687474 703A2F2F 7A6A2D72 6F757465
722F4365 7274456E 726F6C6C 2F7A6A2D 726F7574 65725F43 41253230 53657276
65722E63 72743043 06082B06 01050507 30028637 66696C65 3A2F2F5C 5C7A6A2D
726F7574 65725C43 65727445 6E726F6C 6C5C7A6A 2D726F75 7465725F 43412532
30536572 7665722E 63727430 0D06092A 864886F7 0D010105 05000341 0037500C
D66C236D 2D813702 6C22EFE2 9598DC91 25FE0A3B B0F24869 2C6B9866 BE6B09EF
DE2FDBED 710E04A5 12388B30 2BEBC9D9 881EA210 2C86D23D 25FD9CDF B4
quit
!
!
crypto pki crl url http://zj-router/certsrv/certcrl.crl
!
ip host zj-router 192.168.64.145
!
interface FastEthernet 0/0
ip address 202.101.100.1 255.255.255.0
ip address 192.168.64.199 255.255.255.0 secondary
!
interface FastEthernet 1/0
duplex auto
speed auto
!
interface Null 0
!
!
!
line con 0
line vty 0 4
exec-timeout 0 0
privilege level 15
no login
!
!
end
Ruijie#
```

Monitoring and Maintenance

Ruijie products allow you to query certificate information by using the following command in privileged user mode:

Command	Function
Ruijie# **show crypto pki certificate**	Displays CA root certificate and router certificate configured in the system. When no certificate is configured, there will be no output.

The following shows an example of the **show crypto pki certificate** command output:

```
Ruijie# show crypto pki certificate
%CA certificate info:                            //CA certificate information
Certificate:
Data:
Version: 3 (0x2)                                //X.509v3
Serial Number:                                   //Certificate serial number
7f:ff:bb:39:97:39:b4:81:4b:e1:6b:4f:f9:06:7a:4b
Signature Algorithm: sha1WithRSAEncryption         //Signature algorithm
Issuer: emailAddress=wlcpyjwb@star-net.cn, C=CN, ST=fj, L=fuzhou, O=Red Giant, OU=Department
5, CN=CA Server  //DN name of the issuer
Validity                                      //Certificate validity information
Not Before: Jun 22 05:46:32 2005 GMT      //Effective time in UTC
Not After : Jun 22 05:54:45 2007 GMT      //Time of expiration in UTC
Subject: emailAddress=wlcpyjwb@star-net.cn, C=CN, ST=fj, L=fuzhou, O=Red Giant, OU=Department
5, CN=CA Server
//DN name of certificate subject
Subject Public Key Info:                         //Information about the subject public key
Public Key Algorithm: rsaEncryption       //Public key algorithm: RSA encryption
RSA Public Key: (512 bit)                   //512-bit RSA public key
Modulus (512 bit):
00:be:d1:e8:14:27:7a:30:2b:5e:11:ca:43:fd:2f:
2b:7e:a9:8a:07:96:a2:cf:fe:9d:b7:d3:da:54:c3:
03:4a:a8:44:b3:f0:11:dc:8a:bb:72:53:97:58:b1:
3f:df:6b:8a:9e:5f:46:d3:00:40:2e:24:d3:85:a7:
41:42:55:f7:75
Exponent: 65537 (0x10001)
X509v3 extensions:                            //Certificate extensions
X509v3 Key Usage:                              //Key usage flag
Digital Signature, Non Repudiation, Certificate Sign, CRL Sign           //Including digital
signature, anti-replay, certificate signature, and CRL signature
X509v3 Basic Constraints: critical
CA:TRUE
X509v3 Subject Key Identifier:                 //Subject key identifier
64:46:12:C0:27:A4:9E:01:0C:65:DA:F8:6E:E7:FE:C6:56:EC:AD:D4
X509v3 CRL Distribution Points:                //Information about CRL distribution point
URI:http://zj-router/CertEnroll/CA%20Server.crl
URI:file://\\zj-router\CertEnroll\CA%20Server.crl
1.3.6.1.4.1.311.21.1:
...
Signature Algorithm: sha1WithRSAEncryption          //Signature algorithm
```

```
34:2f:8d:93:68:43:60:7b:68:5f:f0:7e:91:0c:5c:e3:58:98:
7c:53:95:ae:c2:b8:1c:ff:82:a4:ae:95:a8:81:a8:8a:ff:f9:
6f:92:72:3e:fa:6f:84:7d:83:47:93:0f:85:76:48:ae:68:b9:
5a:72:cf:09:50:be:1b:a7:e1:87                       //Certificate signature
%Router certificate info:                  //Information about the router certificate
Certificate:
Data:
Version: 3 (0x2)                              //X.509v3
Serial Number:                                   //Certificate serial number
16:2a:7a:1d:00:00:00:00:00:02
Signature Algorithm: sha1WithRSAEncryption         //Signature algorithm
Issuer: emailAddress=wlcpyjwb@star-net.cn, C=CN, ST=fj, L=fuzhou, O=Red Giant, OU=Department
5, CN=CA Server  //DN name of the issuer
Validity                                      //Certificate validity information
Not Before: Jun 22 05:50:48 2005 GMT      //Effective time in UTC
Not After : Jun 22 06:00:48 2006 GMT      //Time of expiration in UTC
Subject: emailAddress=zhaojun, C=CN, ST=fj, L=fuzhou, O=Red Giant, OU=De partment 5, CN=zhaojun
//DN name of certificate subject
Subject Public Key Info:                       //Information about the subject public key
Public Key Algorithm: rsaEncryption      //Public key algorithm: RSA encryption
RSA Public Key: (2048 bit)                 //2048-bit RSA public key
Modulus (2048 bit):
00:c6:e2:7a:e8:8d:6d:d8:bb:56:a8:9c:03:62:14:
e5:2e:23:e5:a5:26:31:3d:b2:24:65:b1:f2:cc:07:
e3:ef:cc:02:3c:d0:6e:00:8d:fc:ce:3a:b6:45:7a:
cb:a0:87:94:1f:c3:92:43:36:6a:b2:7c:9c:d5:ca:
7e:83:ba:76:49:7f:be:f4:1f:4a:a1:0b:98:22:96:
e2:79:54:a0:ed:1c:62:30:b7:ee:6a:6e:cb:72:e9:
9c:d9:e8:b0:dc:f5:c6:19:8f:2b:2a:85:fa:bf:ff:
08:40:7e:f2:a1:df:d1:8b:ef:68:32:1e:1a:45:fa:
16:de:33:b0:62:90:bd:9c:8e:ec:7c:6e:49:48:75:
e6:5c:ce:b1:8e:1c:80:f3:5b:79:6c:a1:31:b2:a9:
48:37:9f:ed:45:95:85:ba:98:0f:42:c5:78:4c:3d:
a2:45:73:90:3d:0b:1a:7c:53:b5:97:1a:a6:43:2f:
44:54:0f:a1:51:3a:0e:9f:8b:2e:d1:70:cb:36:99:
91:57:d2:b7:9d:7c:ee:07:cf:4a:c7:cd:71:dc:ce:
72:dc:75:a0:03:b2:36:be:8e:af:ca:99:46:03:83:
27:d3:ff:24:1e:4c:0c:21:99:b4:fe:5a:4d:61:b5:
e9:b4:38:dc:59:2c:37:f3:93:02:fc:09:88:02:1b:
d0:45
Exponent: 65537 (0x10001)
X509v3 extensions:                            //Certificate extensions
X509v3 Key Usage: critical          //Key usage flag, which is a key extension
Digital Signature, Non Repudiation       //Including digital signature and anti-replay
X509v3 Extended Key Usage:                      //Extended key usage
```

```
1.3.6.1.5.5.8.2.2
X509v3 Subject Key Identifier:                //Subject key identifier
84:7E:33:A3:91:A5:26:1D:2D:BB:54:65:BF:C7:2A:2A:2E:87:D5:A9
X509v3 Authority Key Identifier:          //Authority key identifier
keyid:64:46:12:C0:27:A4:9E:01:0C:65:DA:F8:6E:E7:FE:C6:56:EC:AD:D4
DirName:/emailAddress=wlcpyjwb@star-net.cn/C=CN/ST=fj/L=fuzhou/O
=Red Giant/OU=Department 5/CN=CA Server
serial:7F:FF:BB:39:97:39:B4:81:4B:E1:6B:4F:F9:06:7A:4B
X509v3 CRL Distribution Points:                //Information about CRL distribution point
URI:http://zj-router/CertEnroll/CA%20Server.crl
URI:file://\\zj-router\CertEnroll\CA%20Server.crl
Authority Information Access:           //Authority information access point
CA Issuers - URI:http://zj-router/CertEnroll/zj-router_CA%20Serv
er.crt
CA Issuers - URI:file://\\zj-router\CertEnroll\zj-router_CA%20Se
rver.crt
Signature Algorithm: sha1WithRSAEncryption          //Signature algorithm
37:50:0c:d6:6c:23:6d:2d:81:37:02:6c:22:ef:e2:95:98:dc:
91:25:fe:0a:3b:b0:f2:48:69:2c:6b:98:66:be:6b:09:ef:de:
2f:db:ed:71:0e:04:a5:12:38:8b:30:2b:eb:c9:d9:88:1e:a2:
10:2c:86:d2:3d:25:fd:9c:df:b4/                    //Certificate signature
Ruijie#
```

Ruijie products allow you to query the CRL information by using the following command in privileged user mode:

Command	Function
Ruijie# show crypto pki crls	Displays CRL information downloaded by the system.

The following shows an example of the **show crypto pki crl** command output:

```
Ruijie# sh crypto pki crls
Certificate Revocation List (CRL):
Version 2 (0x1)                            //CRL version of X.509v2
Signature Algorithm: sha1WithRSAEncryption          //Signature algorithm
Issuer: /emailAddress=wlcpyjwb@star-net.cn/C=CN/ST=fj/L=fuzhou/O=Red Giant/OU=Department
5/CN=CA Server               //DN of the issuer
Last Update: Jun 22 06:10:27 2005 GMT    //Time of last update in UTC
Next Update: Jun 29 18:30:27 2005 GMT  //Time of next update in UTC, namely the expiration
time of CRL
CRL extensions:                         //CRL extensions are shown below
X509v3 Authority Key Identifier:          //Authority key identifier
keyid:64:46:12:C0:27:A4:9E:01:0C:65:DA:F8:6E:E7:FE:C6:56:EC:AD:D4
1.3.6.1.4.1.311.21.1:
...
Revoked Certificates:                      //List of revoked certificates are shown below
Serial Number: 162A7A1D000000000002           //Serial number of the revoked certificate
Revocation Date: Jun 22 06:19:53 2005 GMT          //Revocation date
```

```
CRL entry extensions:                         //CRL entry extensions
X509v3 CRL Reason Code:                     //CRL revocation reason code
Key Compromise                                     //Key compromise
Serial Number: 1635E5E3000000000003
Revocation Date: Jun 22 06:19:53 2005 GMT
CRL entry extensions:
X509v3 CRL Reason Code:
Key Compromise                                     //Key compromise
Signature Algorithm: sha1WithRSAEncryption         //Signature algorithm
5d:a2:ab:07:ff:7e:0e:9a:af:b2:25:11:7f:31:86:aa:21:48:
37:e7:22:99:e3:b2:15:e0:f9:80:63:66:5e:2f:f2:d6:c0:ea:
ef:46:7e:d1:c1:b2:66:0e:0b:d3:74:d1:55:bc:5c:13:46:e8:
56:ec:40:83:7b:1b:75:f2:68:87                       //Signature value
Ruijie#
```

Caution When the **crypto pki revocation-check none** command is used during startup, the existing URL file will not be resolved automatically after startup, and the **show crypto pki crl** command has no output.

Ruijie products allow you to query system debugging information displayed in certificate operations by using the follow0069ng commands in privileged user mode:

Command	Function
Ruijie# **debug crypto pki event**	Displays event tracking information about relevant certificate operations
Ruijie# **debug crypto pki error**	Displays error tracking information about relevant certificate operations

The preceding debugging information will help you diagnose the problems arising during digital certificate configuration and application.

Configuring VPDN

Overview of VPDN

RGOS supports two types of VPDN tunnels: L2TP and PPTP.

Layer Two Tunneling Protocol (L2TP);

Point-to-Point Tunneling Protocol (PPTP).

These two types of VPDN tunneling protocols have their own history. For their configuration and usage, see the following sections. PPTP is commonly used in the Microsoft Windows series products, while L2TP is commonly used in the network devices from such vendors as Cisco. As an industry standard, L2TP is supported by Windows 2000/XP.

Caution

In this chapter, a router refers to the generic route and security gateway unless specially specified.

Configuring PPTP

Overview of PPTP

Point-to-Point-Tunneling Protocol (PPTP) is a network technology that supports multi-protocol VPN. With the PPTP protocol, remote users can dial in the local ISP through Microsoft Windows NT® Workstation, Windows® 95, Windows® 98, Windows® 2000 and other systems with the PPP function enabled, so as to connect and access the corporate network over the Internet securely. Its standard description document is RFC 2637, which is proposed jointly by Microsoft and several industry leading communication device developers. Now it has been recommended to Internet Engineering Task Force(IETF).

The PPTP protocol transmits PPP packets through the tunnel in the IP network. Although it does not modify the PPP protocol in any way, it defines a new PPP packet carrier. By defining the client-server architecture, PPTP divides the functions of the network access server (NAS) and has them implemented by PPTP network server (PNS) and PPTP access concentrator (PAC). The PNS is designed to run on general operating systems. Based on the TCP/IP network, it only requires IP interfaces. The PAC typically has one or more PSTNs, ISDNs or other PPP-enabled physical interfaces.

RGOS now can be used as the PNS, that is, it accepts the PPTP tunnel initiated by the remote client. In this case, the router accepts connection requests from the remote PPTP client and negotiates with the client to establish tunnels.

Note

Now the PNS is only supported on the R26, R36, SecVPN, and RSR series routers but not on the NBR1000.

Configuring the PPTP Server

Configuration Tasks

Configuring a Local Address Pool (Optional)

In order to accept the PPTP connection initiated by the remote client, the PNS must allocate an IP address to the remote client if no IP address is set for it to use internal VPN. Generally, an idle IP address in a specified address pool is allocated to the client. RGOS provides the following commands to configure the local address pool.

Command	Function
Ruijie(config)# **ip local pool** *poolname first-ip* [*last-ip*]	Creates a local address pool.
Ruijie(config)# **no ip local pool** *poolname*	Deletes a specified address pool.

poolname is the name of the local address pool to be created, *first-ip* and *last-ip* are the first and last address in the address range set for the local address pool, respectively.

Configuring User Information (Optional)

To authenticate the remote client that tries to access the local PNS, run the following command.

Command	Function
Ruijie(config)# **username** *user-name* **password** *password*	Configures the user information.
Ruijie(config)# **no username** *user-name*	Deletes the specified user.

user-name is the name of the user who is allowed to dial in, and *password* is the password of the user. The router maintains a local database that contains the user names and passwords.

Configuring VPDN Globally

Enabling/Disabling the VPDN Function

If the user requires the router to accept the PPTP access by the remote client and establish a PPTP tunnel, the VPDN function must be enabled on the router. To enable or disable the VPDN function, run the following command:

Command	Function
Ruijie(config)# **vpdn enable**	Enables the VPDN function.
Ruijie(config)# **no vpdn enable**	Disables the VPDN function.

Note

that if the VPDN function is disabled, all the existing PPTP tunnels and sessions are retained, but new PPTP tunnels and sessions cannot be created.

Setting Source Address of VPDN

RGOS offers the following commands for users to set the (local) source address of the VPDN function. After the source address of VPDN is set, the destination address of the tunnel set for the remote client must match it before a PPTP tunnel can be established properly.

Command	Function
Ruijie(config)# **vpdn source-ip** *ip-address*	Sets the source address of VPDN.
Ruijie(config)# **no vpdn source-ip** *ip-address*	Cancels the set source address of VPDN.

By default, the system does not check whether the destination address in the received tunnel establishment request is a specific value.

Setting Maximum Number of VPDN Sessions

To set the maximum number of sessions allowed by the VPDN server, run the following command. Once specified, the access requests that exceed the maximum value will be denied.

Command	Function
Ruijie(config)# **vpdn session-limit sessions**	Sets the maximum number of VPDN sessions.

Ruijie(config)# **no vpdn session-limit**	Restores the maximum number of VPDN sessions to the default value.

By default, the maximum number of sessions is one configured with this command.

Setting Domain Resolution

RGOS offers the following commands for users to set the domain resolution in VPDN domain authentication. With this command configured, the domain type can be identified.

Command	**Function**
Ruijie(config)# **vpdn domain-delimiter** @/%#-\	Sets the domain delimiter: prefix and suffix.
Ruijie(config)# **no vpdn domain-delimiter**	Cancels the VPDN domain authentication option.

By default, the system does not resolve the domain field.

Enabling Domain Authentication

RGOS offers the following commands for users to set the VPDN domain authentication function.

Command	**Function**
Ruijie(config)# **vpdn authorize domain split**	Enables the domain authentication, and enables the domain split.
Ruijie(config)# **no vpdn authorize domain**	Disables the domain authentication.

By default, the system does not enable the domain authentication.

Setting VPDN Rate Limiting

RGOS offers the following commands for users to limit the rate of creating VPDN sessions, namely, to limit the number of VPDN tunnels allowed to be created at one time.

Command	**Function**
Ruijie(config)# **vpdn limit_rate** *rate_num*	Enables rate limiting. The *rate_num* parameter indicates the number of tunnels allowed to be created, ranging from 5 to 100.
Ruijie(config)# **no vpdn limit_rate**	Disables rate limiting.

By default, the system does not enable rate limiting.

Configuring a Virtual-Template Interface

Setting a Virtual-Template Interface

To set the virtual-template interface, run the following commands.

Command	**Function**
Ruijie(config)# **interface virtual-template** *number*	Creates a specified virtual -template interface.
Ruijie(config)# **no interface virtual-template** *number*	Deletes the specified virtual- template interface.

number is the sequence number of the specified virtual-template interface. The created virtual-template will act as the configuration profile of the virtual-access interface that binds and carries PPTP sessions.

Configuring VPDN Group

Setting VPDN Group

To set a VPDN group, run the followng commands:

Command	Function
Ruijie(config)# **vpdn-group** *name*	Configures a VPDN group.
Ruijie(config)# **no vpdn-group** *name*	Deletes a VPDN group.

name is the name of the VPDN group. Users can access the VPDN group to establish a tunnel.

Setting Tunneling Mode

To set the tunneling mode, run the following commands:

Command	Function
Ruijie (config-vpdn)# **accept-dialin**	Permits the remote client’s access.
Ruijie (config-vpdn)# **no accept-dialin**	Denies the remote client’s access.

If a user wants the local router to perform the PNS function, the user must allow the remote client to dial in.

Setting Tunneling Protocol

To set the tunneling protocol, run the following commands.

Command	Function
Ruijie(config-vpdn-acc-in)# **protocol** {**any** | **l2tp** | **pptp**}	Sets the tunneling protocol.
Ruijie(config-vpdn-acc-in)# **no protocol**	Cancels the set tunneling protocol.

The tunneling mode must be set before the tunneling protocol is set. To make the local router perform the PNS function, the user must run the **protocol pptp** or **protocol any** command.

Setting Virtual Template to Be Used

To set a virtual template used by a VPDN group, run the following commands.

Command	Function
Ruijie(config-vpdn-acc-in)# **virtual-template** *number*	Sets a virtual template to be used.
Ruijie(config-vpdn-acc-in)# **no virtual-template**	Cancels the virtual template in use.

The tunneling mode must be set first before a virtual template used by a VPDN group is set.

Setting the Name of the Remote Peer

If the name of the remote client has been set, this VPDN group is effective only for the remote client that matches the host name. If not, this VPDN group will become the default VPDN group of the system, and can provide the VPDN service for any remote client. If the name of remote client is not configured for any VPDN group, the system will use the first found VPDN group that matches conditions to accept access from a remote dial-in user.

Command	**Function**
Ruijie(config-vpdn)# **terminate-from hostname** *name*	Sets the name of the remote host.
Ruijie(config-vpdn)# **no terminate-from**	Cancels the set name of the remote host.

name indicates the name of the remote host.

Setting Local Name

To set the local name, run the following commands. This name will be sent to the remote peer as a record property.

Command	**Function**
Ruijie(config-vpdn)# **local name** *name*	Sets the local name.
Ruijie(config-vpdn)# **no local** *name*	Cancels the set local name.

name indicates the local name. By default, RGOS uses the name of the router as the local name and sends it to the remote host of the tunnel.

Setting Source Address of VPDN group

To set the source address of a VPDN group, run the following commands. Only when the destination address in the tunnel establishment request sent by the remote client matches it, will the corresponding VPDN group apply.

Command	**Function**
Ruijie(config-vpdn)# **source-ip** *src-ip*	Sets the source address of a VPDN group.
Ruijie(config-vpdn)# **no source-ip**	Cancels the set source address of a VPDN group.

Setting PPTP Flow Control Parameters

To set the PPTP flow control parameters, run the following commands. In general application, the default value of this parameter can be used.

Command	**Function**
Ruijie(config-vpdn)# **pptp flow-control receive-window winsize**	Sets the size of the receive window of the PPTP session. The value range is 1 to 64
Ruijie(config-vpdn)# **no pptp flow-control receive-window**	Cancels the set size of the receive window of the PPTP session and restores to the default value. PAC is 16, and PNS is 64
Ruijie(config-vpdn)# **pptp flow-control** *static-rtt interval*	Sets the static reference time of waiting for ACK on receiving/sending PPTP session packets. The value range is 100 to 5000 (in milliseconds).

Command	Function
Ruijie(config-vpdn)# **no pptp flow-control**	Cancels the set static reference time of waiting for ACK on receiving/sending PPTP session packets, and restore to the default value 1500 milliseconds.

winsize is the size of the receive window of the PPTP session, and *interval* is the static reference time of waiting for an ACK message.

Setting PPTP Tunnel Parameters

To set the PPTP tunnel parameters, run the following commands. In general application, the default value of this parameter can be used.

Command	Function
Ruijie(config-vpdn)# **pptp tunnel echo** *interval*	Sets the time interval at which the PPTP tunnel actively sends echo messages. The value range is 0 to 1000 (in seconds).
Ruijie(config-vpdn)# **no pptp tunnel echo**	Cancels the set interval of sending PPTP echo messages, and restore to the default value 60 seconds.

interval is the time interval at which the PPTP tunnel actively sends ECHO messages. Value **0** indicates that the tunnel does not actively send ECHO messages. The values other than **0** indicates that the tunnel actively sends ECHO messages to detect the tunnel status after it does not receive any valid packet from the remote end of the tunnel within this time interval.

Setting the Supported Domain Name

To set the domain name, run the following commands. After the domain authentication is enabled, this command will take effect. Only the domain matching the content of this command can be identified. If the domain does not match the content of this command, another VPDN group will be used for matching. If no matched group is found, the authentication will fail.

Command	Function
Ruijie(config-vpdn)# **domain** *domain-name* **vrf** *vrf-name*	Sets the authentication domain name and the corresponding VRF instance.
Ruijie(config-vpdn)# **no pool**	Removes the domain setting.

domain-name is the name of a domain, and *vrf-name* is the name of a VRF instance.

Binding a Domain Name to an Address Pool

To bind a domain name to an address pool, run the following command in vpdn-domain configuration mode. The domain name is verified during the process of VPDN tunnel negotiation to obtain the address pool binding information configured with this command. After the PPP negotiation is successful, the specified address pool will be used to assign an address to the peer end of the tunnel. By default, the address is assigned by the address pool configured in the virtual-template interface.

Command	Function
Ruijie(config-vpdn)# **domain** *domain-name* **vrf** *vrf-name* Ruijie(config-vpdn-domain)# **pool** *pool-name*	Sets the address pool bound to the authentication domain name.

Ruijie(config-vpdn)# **no domain** *domain-name*	Removes address pool binding.

domain-name is the name of a domain, *vrf-name* is the name of a VRF instannce and *pool-name* is the name of an address pool.

Setting the DNS Negotiation Address for Binding PPP to the Domain Name

To set the DNS negotiation address of PPP through domain name matching, run the following command in vpdn-domain configuration mode. By default, the DNS address of PPP configured in the virtual-template interface is used for negotiation.

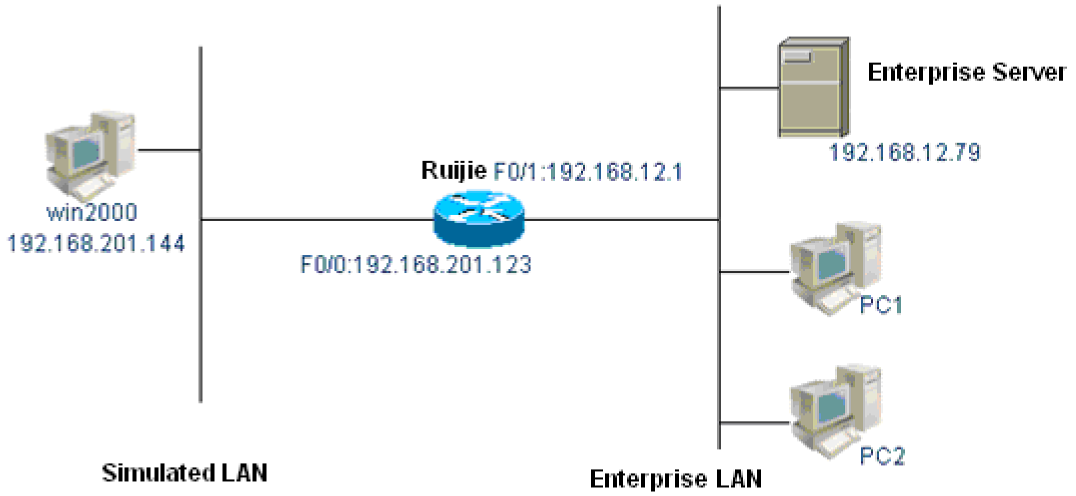
Command	Function
Ruijie(config-vpdn)# **domain** *domain-name* **vrf** *vrf-name* Ruijie(config-vpdn-domain)#**dns** *A.B.C.D A.B.C.D*	Sets the DNS negotiation address of PPP bound to the authentication domain name.
Ruijie(config-vpdn)# **no dns**	Removes DNS binding.

domain-name is the name of a domain, *vrf-name* is the name of a VRF instance and *A.B.C.D* is the address of the DNS.

Configuration Examples

The network topology in this example is shown in the following figure. The Ruijie router is used as the gateway of the corporate LAN, and the PPTP protocol is used to provide the VPDN dial-in service. Use a Windows 2000 PC as the VPDN remote client to create a PPTP tunnel to the router and access the server in the corporate LAN. To reduce testing complexity, use Ethernet to simulate the IP WAN between the remote client and R3660. The IP addresses are configured as shown in the following diagram.

Figure 19 Using a router as the PNS



The configurations of R3660 and Windows 2000 PC are respectively described as follows:

18) Configuration of R3660:

```
Ruijie# show running-config
Building configuration...
Current configuration : 1053 bytes
```

```
!
enable password 1
!
vpdn enable
!
vpdn-group pptp
! Default PPTP VPDN group
accept-dialin
protocol pptp
virtual-template 1
!
username pc password 0 1111
!
ip local pool pptp 1.1.1.2 1.1.1.254
interface FastEthernet 0/0
ip address 192.168.201.123 255.255.255.0
duplex auto
speed auto
!
interface FastEthernet 0/1
ip address 192.168.12.1 255.255.255.0
duplex auto
speed auto
!
interface Virtual-Template 1
ppp authentication pap
ip unnumbered FastEthernet 0/1
peer default ip address pool pptp
!
ip route 0.0.0.0 0.0.0.0 FastEthernet0/0
!
line con 0
session-timeout 0
escape-character 29
line aux 0
session-timeout 0
escape-character 29
password 1
line vty 0
login
terminal-type ANSI
escape-character 29
line vty 1 4
login
escape-character 29
```

```
!
!
end
Ruijie#
```

19) Configuration of the Windows 2000 PC:

```
F:\>ver
Microsoft Windows 2000 [Version 5.00.2195]
F:\>ipconfig /all
Windows 2000 IP Configuration
  Host Name . . . . . . . . . . . . : topding
  Primary DNS Suffix . . . . . . . :
  Node Type . . . . . . . . . . . . : Hybrid
  IP Routing Enabled. . . . . . . . : Yes
  WINS Proxy Enabled. . . . . . . . : No
Ethernet adapter local connection:
  Connection-specific DNS Suffix . :
  Description . . . . . . . . . . . : STAR 901 Family Fast Ethernet
r (ACPI)
  Physical Address. . . . . . . . . : 00-D0-F8-00-68-E5
  DHCP Enabled. . . . . . . . . . . : No
  IP Address. . . . . . . . . . . . : 192.168.201.144
  Subnet Mask . . . . . . . . . . . : 255.255.255.0
  Default Gateway . . . . . . . . . : 192.168.201.123
  DNS Servers . . . . . . . . . . . : 202.101.143.141
  Primary WINS Server . . . . . . . : 192.168.9.7
F:\>route print
===========================================================
Interface List
0x1 ........................... MS TCP Loopback interface
0x2 ...00 d0 f8 00 68 e5 ...... PCI Bus Master Adapter
===========================================================Active Routes:
Network Destination Netmask Gateway  Interface Metric
  0.0.0.0    0.0.0.0 192.168.201.123 192.168.201.144   1
  127.0.0.0   255.0.0.0   127.0.0.1   127.0.0.1     1
192.168.201.0   255.255.255.0 192.168.201.144 192.168.201.144  1
192.168.201.144  255.255.255.255   127.0.0.1   127.0.0.1    1
192.168.201.255  255.255.255.255 192.168.201.144 192.168.201.144   1
224.0.0.0     224.0.0.0 192.168.201.144 192.168.201.144    1
255.255.255.255  255.255.255.255 192.168.201.144 192.168.201.144   1
Default Gateway:  192.168.201.123
===========================================================
Persistent Routes:
None
```

Double-click **Network and Dial-up Connections** on the Windows 2000 PC to create a network connection. Select **Connect to a private network through the Internet** for **Network connection type**, select **Do not dial initial**

connection for **Public network**, and fill in the destination address 192.168.201.123. Name this connection as Vpdnconnect. On the page for setting the properties of Vpdnconnect, use PPTP as the VPDN server type, and define the PAP authentication and optional encryption in the security settings. After you click **Dial**, enter the user name **PC** and password **1111** configured in the router.

Upon completion of configuration, the Windows 2000 PC can access the server, for exmaple, the server with the IP address as 192.168.12.79, in the corporate intranet after dialing in the router, as shown below:

```
F:\>ipconfig /all
Windows 2000 IP Configuration
  Host Name . . . . . . . . . . . . : testpc
  Primary DNS Suffix . . . . . . . :
  Node Type . . . . . . . . . . . . : Hybrid
  IP Routing Enabled. . . . . . . . : Yes
  WINS Proxy Enabled. . . . . . . . : No
Ethernet adapter local connection:
  Connection-specific DNS Suffix . :
  Description . . . . . . . . . . . : STAR 901 Family Fast Ethernet Adapte
r (ACPI)
  Physical Address. . . . . . . . . : 00-D0-F8-00-68-E5
  DHCP Enabled. . . . . . . . . . . : No
  IP Address. . . . . . . . . . . . : 192.168.201.144
  Subnet Mask . . . . . . . . . . . : 255.255.255.0
  Default Gateway . . . . . . . . . :
  DNS Servers . . . . . . . . . . . : 202.101.143.141
  Primary WINS Server . . . . . . . : 192.168.9.7
PPP adapter vpn_RGOS:
  Connection-specific DNS Suffix . :
  Description . . . . . . . . . . . : WAN (PPP/SLIP) Interface
  Physical Address. . . . . . . . . : 00-53-45-00-00-00
  DHCP Enabled. . . . . . . . . . . : No
  IP Address. . . . . . . . . . . . : 1.1.1.4
  Subnet Mask . . . . . . . . . . . : 255.255.255.255
  Default Gateway . . . . . . . . . : 192.168.12.1
  DNS Servers . . . . . . . . . . . :
F:\>
F:\>route print
===================================================================
Interface List
0x1 ........................... MS TCP Loopback interface
0x2 ...00 d0 f8 00 68 e5 ...... PCI Bus Master Adapter
0x30000004 ...00 53 45 00 00 00 ...... WAN (PPP/SLIP) Interface
===================================================================
Active Routes:
Network Destination  Netmask   Gateway  Interface Metric
   0.0.0.0   0.0.0.0  192.168.12.1   1.1.1.4  1
  1.1.1.4 255.255.255.255   127.0.0.1   127.0.0.1  1
 1.255.255.255 255.255.255.255   1.1.1.4   1.1.1.4  1
  127.0.0.0   255.0.0.0   127.0.0.1   127.0.0.1  1
 192.168.201.0 255.255.255.0 192.168.201.144 192.168.201.144  1
 192.168.201.123 255.255.255.255 192.168.201.144 192.168.201.144  1
 192.168.201.144 255.255.255.255   127.0.0.1   127.0.0.1  1
 192.168.201.200 255.255.255.255 192.168.201.100 192.168.201.144  1
 192.168.201.255 255.255.255.255 192.168.201.144 192.168.201.144  1
  224.0.0.0   224.0.0.0   1.1.1.4    1.1.1.4  1
  224.0.0.0   224.0.0.0 192.168.201.144 192.168.201.144  1
 255.255.255.255 255.255.255.255 192.168.201.144 192.168.201.144 1
Default Gateway:   192.168.12.1
===================================================================
Persistent Routes:
 None
F:\>
F:\>ping 192.168.12.1
Pinging 192.168.12.1 with 32 bytes of data:
Reply from 192.168.12.1: bytes=32 time<10ms TTL=255
Reply from 192.168.12.1: bytes=32 time<10ms TTL=255
Reply from 192.168.12.1: bytes=32 time<10ms TTL=255
Reply from 192.168.12.1: bytes=32 time<10ms TTL=255
Ping statistics for 192.168.12.1:
 Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
 Minimum = 0ms, Maximum = 0ms, Average = 0ms
F:\>ping 192.168.12.79
Pinging 192.168.12.79 with 32 bytes of data:
Reply from 192.168.12.79: bytes=32 time=10ms TTL=127
Reply from 192.168.12.79: bytes=32 time<10ms TTL=127
Reply from 192.168.12.79: bytes=32 time<10ms TTL=127
Reply from 192.168.12.79: bytes=32 time<10ms TTL=127
Ping statistics for 192.168.12.79:
 Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
 Minimum = 0ms, Maximum = 10ms, Average = 2ms
```

The routing information on the router is shown below:

```
Ruijie#sh ip route
Codes: C - connected, S - static, R - RIP
  O - OSPF, IA - OSPF inter area
  E1 - OSPF external type 1, E2 - OSPF external type 2
Gateway of last resort is 0.0.0.0 to network 0.0.0.0
  1.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C  1.1.1.0/24 is directly connected, Virtual-Access1
```

```
C  1.1.1.4/32 is directly connected, Virtual-Access1
C 192.168.12.0/24 is directly connected, FastEthernet0/1
C 192.168.201.0/24 is directly connected, FastEthernet0/0
S* 0.0.0.0/0 is directly connected, FastEthernet0/0
Ruijie#
```

Note

that in order to enable remote VPDN users to access the intranet servers, routes to these users must be configured on the servers. Generally, you can simply set the default gateway of these servers to the internal gateway address of the router, which is 192.168.12.1 in this example.

Monitoring and Maintaining PPTP

Monitoring PPTP

To query the information about currently created tunnels and remote dial-in users, run the **show vpdn** command.

Command	Function
Ruijie# **show vpdn tunnel**	Displays information about all the existing VPDN tunnels.
Ruijie# **show vpdn session**	Displays information about all the existing VPDN sessions.
Ruijie# **show vpdn**	Displays information about all the existing VPDN tunnels and sessions.

In the configuration example, the following information can be viewed:

```
Ruijie# sh vpdn
%No active L2TP tunnels
PPTP Tunnel and Session Information Total tunnels 1 sessions 1
LocID Remote Name  State  Remote Address Port Sessions
1      estbed  192.168.201.144 1436 1
LocID RemID TunID Intf Username  State  Last Chg
1  49152 1  Vi1  pc   connected 00:31:33
Ruijie#
```

Information about the L2TP and PPTP tunnels and sessions are displayed by category. The tunnel type and statistical values are displayed first, and all the tunnel and session information is displayed later . Statistical values of the tunnels and sessions: Total tunnels 1 sessions 1. The tunnel information includes tunnel ID (LocID), remote host name (Remote Name), tunnel state (State), IP address of the remote host (Remote Address), TCP port number of the PPTP tunnel (Port), number of sessions in this tunnel (Sessions). The session information about the local call ID (LocID), remote call ID (RemID), the ID of the tunnel to which this session belongs (TunID), name of the Virtual-Access interface used by this session (Intf), user name (Username), session state (State), and the time of last state change (Last Chg).

To view detailed tunnel information, run the **show vpdn tunnel pptp** *locid* command. In this example, the following information can be viewed:

```
R3660# show vpdn tunnel pptp 1
PPTP tunnel id 1 is up, remote id is 0, 1 active session
Tunnel state is estbed
Remote tunnel name is
Internet Address 192.168.201.144, port 1436
Local tunnel name is
Internet Address 192.168.201.123
```

The command output contains the tunnel status, name of the peer user, peer IP address, local host name, and local IP address.

Maintaining PPTP

RGOS provides the **clear vpdn** command to clear the specified tunnel and all its sessions.

Command	Function
Ruijie# **clear vpdn tunnel** [[**l2tp** | **pptp**] [*remote name*]]	Clears all the tunnels or the tunnel of the specified type and with the specified remote host name, and all their or its sessions.

remote name is the remote host name for which the tunnel should be cleared. In the configuration example, if **clear vpdn tunnel pptp** or **clear vpdn tunnel** is used after a tunnel is created, the session and the tunnel will be cleared. The command output is as follows:

```
Ruijie# show vpdn
%No active L2TP tunnels
%No active PPTP tunnels
```

During network debugging, you can run the **debug vpdn** command to track the establishment process of PPTP tunnels and sessions. In addition, the debugging information obtained by using the **debug ppp** command is extremely important for tracking call failures. For details, refer to description of the PPP protocol.

Command	Function
Ruijie# **debug vpdn** { **error** | **event** | **packet** }	Displays the debugging information during creation and use of the VPDN tunnel and session on the configuration terminal. **Error** indicates error information, **Event** indicates a general event, and **Packet** indicates the content in the control packet.

Note The debugging information may vary slightly with the RGOS software version.

During creation of PPTP tunnels and sessions, the **debug vpdn event** command outputs the following information:

```
VPDN: Pptp recv start-control-connection-request from host 192.168.200.114
PPTP: New tunnel socket id =9
VPDN: Pptp get tunnel info for 192.168.200.114 ok!
VPDN: Pptp send start-control-connection-reply, ok
VPDN: Pptp tunnel id 0 state change: idle --> estbed
PPTP: Add send-echo-request timer, interval = 60
```

```
VDPN: Pptp tunnel id 0 recv outgoing-call-request!
Pptp: Tunnel to 192.168.200.114 get config para. from vpdn-group pptp!
VPDN: Must process using ACCEPT_DIALIN parameters
Pptp: Session va0 get config para. from vpdn-group pptp!
VPDN: Pptp session va0 state change: idle --> connected
PPTP: Receive outcall request,process ok!assign local call id = 1
VPDN: Pptp tunnel id 0 send out-call reply
%LINK CHANGED: Interface virtual-access 0, changed state to up
VPDN: Pptp tunnel to 192.168.200.114 peer callid 1 recv set-linkinfo
VPDN: Pptp tunnel to 192.168.200.114 peer callid 1 recv set-linkinfo
%LINE PROTOCOL CHANGE: Interface virtual-access 0, changed state to UP
```

During creation of PPTP tunnels and tunnels, the **debug vpdn packet** command outputs the following information:

```
PPTP: I Start-Control-Connection-Request len 156 Magic Cookie 0x1A2B3C4D
   Protocol Version 0x100
   Framing Type 0x1
   Bearer Type 0x1
   Maximum Channels 0x0
   Firmware Revision 0x893
   Host Name:
   Vendor String: Microsoft Windows NT
PPTP: O Start-Control-Connection-Reply len 156 Magic Cookie 0x1A2B3C4D
   Protocol Version 0x100
   Framing Type 0x2
   Bearer Type 0x3
   Maximum Channels 0x0
   Firmware Revision 0x100
   Host Name: Dingjs
   Vendor String: Ret-Giant Network Operating System
PPTP: I Outgoing-Call-Request len 168 Magic Cookie 0x1A2B3C4D
   Call Id 0x4000
   Call Serial Number 0x96A5
   Min BPS 0x12C
   Max BPS 0x5F5E100
   Bearer Type 0x3
   Framing Type 0x3
   Rec Window Size 0x40
   Proc Delay 0x0
   Phone Number Length 0x0
   Phone Number:
   Subaddress:
PPTP: O Outgoing-Call-Reply len 32 Magic Cookie 0x1A2B3C4D
   Call Id 0x1
   Peer Call Id 0x4000
   Result Code 0x1
   Error Code 0x0
   Cause Code 0x0
   Connect Speed 0xFA00
   Rec Window Size 0x10
   Physical Channel Id 0x0
PPTP: I Set-Link-Info len 24 Magic Cookie 0x1A2B3C4D
   Peer Call Id 0x1
   Send ACCM 0xFFFFFFFF
   Recv ACCM 0xFFFFFFFF
%UPDOWN: Interface Virtual-Access1, changed state to up
Vi1 VPDN PROCESS Into tunnel: Sending 54 byte pak
Vi1 VPDN PROCESS Into tunnel: Sending 64 byte pak
Vi1 VPDN PROCESS Into tunnel: Sending 50 byte pak
PPTP: I Set-Link-Info len 24 Magic Cookie 0x1A2B3C4D
   Peer Call Id 0x1
   Send ACCM 0xFFFFFFFF
   Recv ACCM 0xFFFFFFFF
Vi1 VPDN PROCESS Into tunnel: Sending 45 byte pak
Vi1 VPDN PROCESS Into tunnel: Sending 46 byte pak
Vi1 VPDN PROCESS Into tunnel: Sending 187 byte pak
Vi1 VPDN PROCESS Into tunnel: Sending 56 byte pak
Vi1 VPDN PROCESS Into tunnel: Sending 64 byte pak
Vi1 VPDN PROCESS Into tunnel: Sending 50 byte pak
Vi1 VPDN PROCESS Into tunnel: Sending 50 byte pak
Vi1 VPDN PROCESS Into tunnel: Sending 52 byte pak
```

If the physical connection with the client is interrupted, output of the **debug vpdn error** command is as follows:

```
VPDN: PPTP session Virtual-Access1 wait pak ack timeout(wait seq=37, ack=36), de
crease send window to half of current = 33!
VPDN: PPTP session Virtual-Access1 adjust ATO to 220 ms!
VPDN: PPTP session Virtual-Access1 wait pak ack timeout(wait seq=38, ack=36), de
crease send window to half of current = 16!
VPDN: PPTP session Virtual-Access1 adjust ATO to 280 ms!
VPDN: PPTP session Virtual-Access1 wait pak ack timeout(wait seq=39, ack=36), de
crease send window to half of current = 8!
VPDN: PPTP session Virtual-Access1 adjust ATO to 400 ms!
VPDN: Pptp EGRE encap fail, err=-4!
VPDN: PPTP session Virtual-Access1 wait pak ack timeout(wait seq=40, ack=36), de
crease send window to half of current = 4!
VPDN: PPTP session Virtual-Access1 adjust ATO to 640 ms!
```

FAQs

The following gives an FAQ about VPDN access over PPTP.

Suppose a PPTP dial-up connection is created on a Windows 2000 PC.

When setting the properties of the new dial-up connection, set **Security measure options** to **Advanced (user-defined setting) (D)**, set **Data encryption** to **Optional encryption (connect even if no encryption available)** on the **Setting** tab page, and allow PAP, CHAP and MS-CHAP authentication according to the authentication type configured in virtual-template that RGOS uses for PPTP dial-in. Additionally, **VPN server type** on the **Network** tab page is set to **Automatic**. After this setting, the Windows 2000 PC will try to create an L2TP tunnel before it creates a PPTP tunnel. This takes a long time. Therefore, you can set **VPN server type** to **Point-to-Point Tunneling Protocol (PPTP)**. As a result, a PPTP tunnel is directly created without an attempt to creating an L2TP tunnel.

When the RGOS router is located behind other firewalls, the TCP port 1723 of the firewall must be enabled.

Configuring L2TP

Overview

The Layer 2 Tunnel Protocol (L2TP), as specified in RFC 2661, is a standard tunneling protocol that Internet Engineering Task Force (IETF) proposes by combining two existing tunneling protocols, namely, Cisco Layer 2 Forwarding (L2F) protocol and Microsoft Point-to-Point Tunneling Protocol (PPTP).

L2TP, an extension of the Point-to-Point Protocol (PPP), implements user authentication and data transmission using PPP. Different from PPTP, L2TP uses UDP as the transmission protocol for control and data messages.

L2TP is also an important and effective way to implement VPN. VPN allows network users to access the enterprise intranet more conveniently and securely, no matter whether the users access the network in dial-up mode or in other modes.

RGOS supports L2TP tunnels in two modes.

- L2TP tunnel initiated by the local client: In this mode, the router acts as the L2TP client and actively initiates negotiation with the L2TP server to establish a tunnel.
- L2TP tunnel initiated by the remote client: In this mode, the router accepts a connection request from the remote L2TP client and negotiates with it to establish a tunnel.

Note Both modes are supported on the R26, R36, and SecVPN platforms, but the NBR platform supports only L2TP tunnel initiated by the local client.

Initiation by the Local Client

Configuration Task List

- Creating and configuring an L2TP-class interface (optional)
- Creating and configuring a pseudowire-class interface (optional)
- Creating and configuring a virtual-ppp interface (mandatory)

Creating and Configuring an L2TP-class Interface

This is an optional step for establishing an L2TP tunnel initiated by the local client. In this step, you can set the parameters for the L2TP control connection. The operations of configuring an L2TP-class interface include:

- Setting an L2TP-class unit
- Setting time for the L2TP control connection
- Setting authentication for the L2TP control connection
- Setting maintenance and update for the L2TP control connection

Setting an L2TP-class Unit

Use the following commands to set an L2TP-class unit for setting parameters for the L2TP control connection.

Command	Function
Ruijie(config)# **l2tp-class** *l2tp-class-name*	Configures or creates an L2TP-class interface of the specified name.
Ruijie(config)# **no l2tp-class** *l2tp-class-name*	Deletes an L2TP-class interface of the specified name.

l2TP-class-name is the name of the created or set L2TP-class unit. The L2TP-class interface created here can be referenced by the pseudowire-class interface by name.

Setting Time for the L2TP Control Connection

Use the following commands to set time for the L2TP control connection.

Command	Function
Ruijie(config-l2tp-class)# **receive-window** *size*	Sets the size of the receiving window of the control connection.
Ruijie(config-l2tp-class)# **no receive-window**	Restores the default size of the receiving window of the control connection.
Ruijie(config-l2tp-class)# **retransmit** {**initial** {**retries** *initial-retries* | **timeout** {**max** | **min**} *initial-timeout*} | **retries** *retries* | **timeout** {**max** | **min**} *timeout*}	Sets retransmission of the control connection.
Ruijie(config-l2tp-class)# **no retransmit** { **initial** {**retries** | **timeout** {**max** | **min**} }| **retries** | **timeout** {**max** | **min**} }	Restores default retransmission setting of the control connection.
Ruijie(config-l2tp-class)# **timeout setup** *seconds*	Sets the timeout period for establishing a control connection.
Ruijie(config-l2tp-class)# **no timeout setup**	Restores the default timeout period for establishing a control connection.

size is the size of the receiving window, and the default value is 8.

initial-retries is the number of SCCRQ retransmission times, and the default value is 2.

initial-timeout is the interval of SCCRQ retransmission. The default minimum interval is 1 second, and the default maximum interval is 8 seconds.

retries is the number of retransmission times of control messages, and the default value is 5.

timeout is the interval of control message retransmission. The default minimum interval is 1 second, and the default maximum interval is 8 seconds.

seconds is the upper limit of time for establishing a control connection (tunnel), and the default value is 120 seconds.

Setting Authentication for the L2TP Control Connection

Use the following commands to set authentication for the L2TP control connection (tunnel).

Command	Function
Ruijie(config-l2tp-class)# **authentication**	Enables authentication.
Ruijie(config-l2tp-class)# **no authentication**	Disables authentication.
Ruijie(config-l2tp-class)# **no hostname**	Uses the default local host name.
Ruijie(config-l2tp-class)# **hostname** *host-name*	Sets the local host name corresponding to this control connection.
Ruijie(config-l2tp-class)# **password** *pass-words*	Sets the tunnel password.
Ruijie(config-l2tp-class)# **no password**	Cancels the tunnel password.

RGOS does not require tunnel authentication by default, but uses the name of the router as the local host name. If tunnel authentication is required, both ends must use the same tunnel password. *host-name* is the local host name set by users, and *pass-words* is the password used for tunnel authentication.

Setting Maintenance and Update for the L2TP Control Connection

Use the following commands to set maintenance and update for the control connection (tunnel).

Command	Function
Ruijie(config-l2tp-class)# **hello** *interval*	Sets the interval of sending Hello messages.
Ruijie(config-l2tp-class)# **no hello**	Restores the default interval of sending Hello messages.

Here, *interval* is the interval of sending Hello messages. Its default value is 60 seconds.

Creating and Configuring a Pseudowire-class Interface

This is an optional step for establishing an L2TP tunnel initiated by the local client. In this step, you can set L2TP data transmission parameters. The operations of setting a pseudowire-class interface include:

- Setting a pseudowire-class unit
- Setting the encapsulation mode for L2TP data transmission
- Setting IP parameters for L2TP data transmission
- Setting the L2TP control connection

Setting a Pseudowire-class Unit

Use the following commands to set a pseudowire-class unit for seting L2TP data transmission parameters.

Command	Function
Ruijie(config)# **pseudowire-class** *pseudowire-class-name*	Creates or configures a pseudowire-class interface of the specified name.
Ruijie(config)# **no pseudowire-class** *pseudowire-class-name*	Deletes the pseudowire-class interface of the specified name.

pseudowire-class-name is the name of the created or set pseudowire-class unit. Here, the created pseudowire-class interface can be referenced by the peudowire rule of the virtual-ppp interface by name.

Setting the Encapsulation Mode for L2TP Data Transmission

Use the following command to set the encapsulation mode for L2TP data transmission.

Command	Function
Ruijie (config-pw-class)# **encapsulation l2tpv2**	Sets the encapsulation mode for L2TP data transmission.

Note that once the encapsulation mode is set for data transmission in L2TP channels, it cannot be changed. If a user needs to set L2TP data transmission parameters on the peudowire-class interface, the user must first set the encapsulation mode for L2TP data transmission.

Setting IP Parameters for L2TP Data Transmission

Use the following commands to set IP parameters for L2TP data transmission.

Command	Function
Ruijie (config-pw-class)# **ip dfbit set**	Disables channel data fragmentation.
Ruijie (config-pw-class)# **no ip dfbit set**	Enables channel data fragmentation.
Ruijie (config-pw-class)# **ip ttl** *ttl-value*	Sets TTL for the IP header of the channel.
Ruijie (config-pw-class)# **no ip ttl**	Restores the default TTL.
Ruijie (config-pw-class)# **ip local interface** *interface-name*	Specifies the local interface (address) of the channel.
Ruijie(config-pw-class)# **no ip local interface** *interface-name*	Cancels the specified local interface (address) of the channel.

Setting IP parameters for L2TP data transmission actually means setting the IP header of the UDP data that carries L2TP. The system allows channel data fragmentation by default. The default TTL is 255, and the system will set the nearest local address (interface) in the route for it based on the specified peer address.

Setting the L2TP Control Connection

Use the following commands to set the L2TP control connection.

Command	Function
Ruijie(config-pw-class)# **protocol l2tpv2** [*l2tp-class-name*]	Sets the L2TP control connection parameter.
Ruijie(config-pw-class)# **no protocol**	Use the default control connection parameter.

Here, **l2tpv2** creates a control connection in compliance with the L2TP protocol specified in the RFC 2661. *l2TP-class-name* is set to an existing L2TP-class interface to limit the control connection parameter. If no L2TP-class interface is available, the default L2TP control connection parameter is used.

Creating and Configuring a Virtual-ppp Interface

This is a mandatory step for establishing an L2TP tunnel initiated by the local client. A specified L2TP session will be created in this step. The operations of setting a virtual-ppp interface include:

- Setting a virtual-ppp interface

- Setting the IP address
- Setting authentication
- Setting the pseudowire rule

For information about setting the IP address and authentication parameter, see related sections in the interface configuration guide.

Setting a Virtual-ppp Interface

Use the following commands to set a virtual-ppp interface for establishing an L2TP session.

Command	**Function**
Ruijie(config)# **interface virtual-ppp** *number*	Creates or configures a specified virtual-ppp interface.
Ruijie(config)# **no interface virtual-ppp** *number*	Deletes the specified virtual-ppp interface.

number is the name of the specified virtual-ppp interface. The created virtual-ppp interface will be used to create and bind an L2TP session.

Setting the Pseudowire Rule

Use the following commands to set the pseudowire rule on the virtual-ppp interface for establishing an L2TP session.

Command	**Function**
Ruijie (config-if)# **pseudowire** *peer-ip-address vcid* {**encapsulation l2tpv2** [**pw-class** *pw-class-name*] | **pw-class** *pw-class-name* }	Sets the pseudowire rule.
Ruijie (config-if)# **no pseudowire**	Deletes the pseudowire rule.

Once the pseudowire rule is set on a virtual-ppp interface, the virtual-ppp interface will automatically attempt to establish an L2TP session with the specified LNS. If a failure occurs, the virtual-ppp interface will attempt to establish an L2TP session 10 seconds later again. Here, *peer-ip-address* is the address of the remote LNS, *vcid* is the global ID, and *pw-class-name* is the name of the referenced pseudowire-class interface.

You can set an L2TP session with the specified LNS name when the DNS service is enabled. Our products support only the DNS client service, and the name of specified LNS must be registered on the DNS server.

Command	**Function**
Ruijie (config-if)# **pseudowire** *peer-ip-address vcid* {**encapsulation l2tpv2** [**pw-class** *pw-class-name*] | **pw-class** *pw-class-name* }	Sets the pseudowire rule.
Ruijie (config-if)# **no pseudowire**	Deletes the pseudowire rule.

Once the pseudowire rule is set on a virtual-ppp interface, the virtual-ppp interface will automatically attempt to establish an L2TP session with the specified LNS. If a failure occurs, the virtual-ppp interface will attempt to establish an L2TP session 10 seconds later again. Here, *peer-hostname* is the hostname of the remote LNS. Ruijie DNS will convert this hostname to a specific IP address (note that the hostname must be registered on the DNS server). *vcid* is the global ID, and *pw-class-name* is the name of the referenced pseudowire-class interface.

Setting the VRF Attribute

Use the following commands to set the VRF attribute on the virtual-ppp interface for establishing an L2TP session.

Command	Function
Ruijie(config-Virtual-ppp 1)# **vpdn vrf** *vrf-name*	Sets the name of the VRF to which L2TP tunnel packets belong.
Ruijie(config-Virtual-ppp 1)#**no vpdn vrf**	Deletes the VRF attribute configuration.

The command for setting the VRF attribute is generally used together with the **ip vrf forward** command of an interface. Once the VRF attribute is configured on the virtual-ppp interface, the encapsulated packets will be sent to the specified VRF. If the VRF of the interface is different from that of the tunnel, the VRF attribute of packets changes before and after encapsulation, implementing VRF spanning.

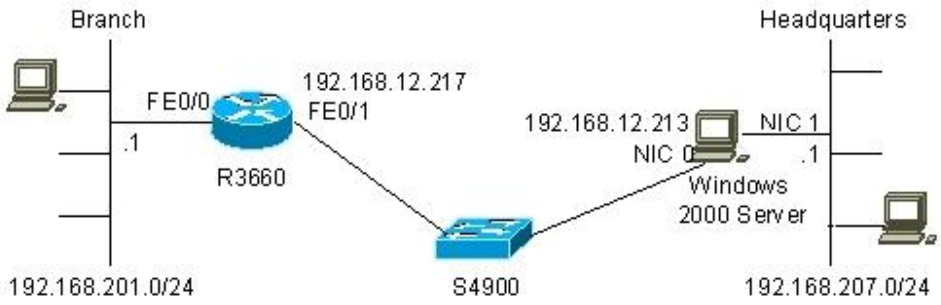
Configuration Examples

Two configuration examples are given below. In one configuration example, The Windows 2000 server is used as the remote L2TP server, and the tunnel authentication is not performed due to limitations of the Windows 2000 server. In the other configuration example, Cisco 2620 is used as the L2TP server, and tunnel authentication is performed.

Establishing a Tunnel with the Windows 2000 Server

Figure shows the networking topology of the L2TP tunnel established by using Ruijie router and the Windows 2000 server.

Figure 20 Networking topology of the L2TP tunnel established by using the Windows 2000 server (LNS)



The configurations of the R3660 and Windows 2000 server are respectively described as follows:

20) R3660 configuration:

```
R3660# show running-config
Building configuration...
Current configuration : 868 bytes
!
hostname R3660
access-list 101 permit ip any 192.168.207.0 0.0.0.255
access-list 102 deny   ip any 192.168.207.0 0.0.0.255
access-list 102 permit ip any any
!
l2tp-class l2x
```

```
hostname branch
!
pseudowire-class pw
encapsulation l2tpv2
protocol l2tpv2 l2x
ip local interface FastEthernet 0/1
!
!
!
!
!
!
interface FastEthernet 0/0
ip address 192.168.201.1 255.255.255.0
ip nat inside
duplex auto
speed auto
!
interface FastEthernet 0/1
ip address 192.168.12.217 255.255.255.0
ip nat outside
duplex auto
speed auto
!
interface Null 0
!
interface Virtual-ppp 1
pseudowire 192.168.12.213 12 pw-class pw
ppp pap sent-username rgnos password 7 072C04211A01
ip mtu 1460
ip address negotiate
ip nat outside
!
ip nat inside source list 102 interface FastEthernet0/1 overload
ip nat inside source list 101 interface Virtual-PPP1 overload
ip route 0.0.0.0 0.0.0.0 FastEthernet 0/1 192.168.12.1
ip route 192.168.207.0 255.255.255.0 Virtual-ppp 1
!
line con 0
line aux 0
line vty 0 4
!
!
end
```

In this configuration, the intranet of the branch shares (using the NAT function) the L2TP tunnel that the interface virtual-ppp 1 establishes with the headquarters, to access the intranet of the headquarters. The intranet of the branch shares (using the NAT function) the WAN interface FastEthernet 0/1 to access the Internet. Distribution of such data streams is controlled by using the access control list (ACL).

21) Configuration of the Windows 2000 server:

```
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.
C:\>ipconfig /all
Windows 2000 IP Configuration
       Host Name . . . . . . . . . . . . : BLIZZARD
       Primary DNS Suffix  . . . . . . . :
       Node Type . . . . . . . . . . . . : Broadcast
       IP Routing Enabled. . . . . . . . : Yes
       WINS Proxy Enabled. . . . . . . . : No
Ethernet adapter local connection 2:
       Connection-specific DNS Suffix  . :
       Description . . . . . . . . . . . : NE2000 Compatible
       Physical Address. . . . . . . . . : 00-10-88-01-A5-C3
       DHCP Enabled. . . . . . . . . . . : No
       IP Address. . . . . . . . . . . . : 192.168.12.213
       Subnet Mask . . . . . . . . . . . : 255.255.255.0
       Default Gateway . . . . . . . . . : 192.168.12.1
       DNS Servers . . . . . . . . . . . : 202.101.143.141
PPP adapter RAS Server (Dial In) Interface:
       Connection-specific DNS Suffix  . :
       Description . . . . . . . . . . . : WAN (PPP/SLIP) Interface
       Physical Address. . . . . . . . . : 00-53-45-00-00-00
       DHCP Enabled. . . . . . . . . . . : No
       IP Address. . . . . . . . . . . . : 192.168.103.2
       Subnet Mask . . . . . . . . . . . : 255.255.255.255
       Default Gateway . . . . . . . . . :
       DNS Servers . . . . . . . . . . . :
C:\>route print
======================================================================
Interface List
0x1 ........................... MS TCP Loopback interface
0x1000002 ...00 53 45 00 00 00 ...... WAN (PPP/SLIP) Interface
0x1000003 ...00 10 88 01 a5 c3 ...... Novell 2000 Adapter.
======================================================================
Active Routes:
Network Destination       Netmask         Gateway       Interface  Metric
         0.0.0.0         0.0.0.0     192.168.12.1  192.168.12.213     1
       127.0.0.0       255.0.0.0        127.0.0.1       127.0.0.1     1
    192.168.12.0   255.255.255.0   192.168.12.213  192.168.12.213     1
  192.168.12.213 255.255.255.255        127.0.0.1       127.0.0.1     1
```

```
   192.168.12.217  255.255.255.255   192.168.12.213  192.168.12.213      1
   192.168.12.255  255.255.255.255   192.168.12.213  192.168.12.213      1
    192.168.103.2  255.255.255.255        127.0.0.1       127.0.0.1      1
    192.168.103.6  255.255.255.255    192.168.103.2    192.168.103.2     1
        224.0.0.0        224.0.0.0   192.168.12.213   192.168.12.213     1
  255.255.255.255  255.255.255.255   192.168.12.213   192.168.12.213     1
Default Gateway:     192.168.12.1
======================================================================
Persistent Routes:
  None
C:\>
```

Note that the routing and remote access function must be enabled on the Windows 2000 server to accept the remote VPDN access. Set the access control policies (including the ACL, authentication type, IP address allocation policies, and encryption method). Default values are used here. Then, in **Network and Dial-up Connections**, click **New connection** and **Accept incoming connections** to accept the access from the remote L2TP, and set which users' L2TP access requests are accepted.

Note

On Windows 2000/XP, L2TP is bound with IPSec/IKE, which undoubtedly increases the workload of network administrators, because most L2TP clients (such as network devices from Cisco and Quidway) do not bind L2TP to IPSec/IKE. Establishing an L2TP tunnel by using a Windows 2000/XP PC is also difficult due to such binding. Fortunately, network administrators can cancel the binding by modifying the registry on Windows 2000/XP. To do so, choose **Start** > **Run**, and then enter **regedit** to open the registry editor. Find the directory **HKEY_LOCAL_MACHINE/ SYSTEM / CurrentControlSet / Services / RasMan /Parameters**. Create a double-byte value named **ProhibitIpSec** and set it to **1**. Press **F5** to refresh the registry, and finally, restart the Windows 2000. Then, network administrators do not need to consider the complicated IPSec/IKE setting when using L2TP.

L2TP tunnels of Windows 2000/XP mentioned in this document are not bound with IPSec/IKE unless otherwise specified.

The following shows how the host "DENGL-NECBOOK" of the branch accesses the Internet and the intranet of the headquarters.

```
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.
C:\WINNT\system32>ipconfig /all
Windows 2000 IP Configuration
       Host Name . . . . . . . . . . . . : DENGL-NECBOOK
       Primary DNS Suffix  . . . . . . . :
       Node Type . . . . . . . . . . . . : Hybrid
       IP Routing Enabled. . . . . . . . : No
       WINS Proxy Enabled. . . . . . . . : No
Ethernet adapter local connection:
       Connection-specific DNS Suffix  . :
       Description . . . . . .: Realtek RTL8139(A)-based PCI Fast Ethernet Adapter
       Physical Address. . . . . . . . . : 00-10-60-75-BD-7A
       DHCP Enabled. . . . . . . . . . . : No
       IP Address. . . . . . . . . . . . : 192.168.201.78
       Subnet Mask . . . . . . . . . . . : 255.255.255.0
       Default Gateway . . . . . . . . . : 192.168.201.1
       DNS Servers . . . . . . . . . . . : 202.101.143.141
                                     202.101.98.55
C:\WINNT\system32>route print
======================================================================
Interface List
0x1 ........................... MS TCP Loopback interface
0x1000003 ...00 10 60 75 bd 7a ...... NDIS 5.0 driver
======================================================================
Active Routes:
Network Destination       Netmask        Gateway      Interface  Metric
        0.0.0.0          0.0.0.0    192.168.201.1  192.168.201.78    1
       127.0.0.0        255.0.0.0       127.0.0.1       127.0.0.1    1
   192.168.201.64  255.255.255.192  192.168.201.78  192.168.201.78    1
   192.168.201.78  255.255.255.255       127.0.0.1       127.0.0.1    1
  192.168.201.255  255.255.255.255  192.168.201.78  192.168.201.78    1
        224.0.0.0        224.0.0.0  192.168.201.78  192.168.201.78    1
  255.255.255.255  255.255.255.255  192.168.201.78   192.168.201.78   1
Default Gateway:   192.168.201.1
======================================================================
Persistent Routes:
  Network Address        Netmask  Gateway Address  Metric
     192.168.2.0   255.255.255.0     192.168.1.1      1
     192.168.9.0   255.255.255.0    192.168.12.1      1
     61.154.22.0   255.255.255.0    192.168.12.1      1
C:\WINNT\system32>ping 192.168.12.1
Pinging 192.168.12.1 with 32 bytes of data:
Reply from 192.168.12.1: bytes=32 time=50ms TTL=254
Reply from 192.168.12.1: bytes=32 time=20ms TTL=254
Reply from 192.168.12.1: bytes=32 time<10ms TTL=254
Reply from 192.168.12.1: bytes=32 time=60ms TTL=254
Ping statistics for 192.168.12.1:
   Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
   Minimum = 0ms, Maximum =  60ms, Average =  32ms
C:\WINNT\system32>ping 192.168.103.2
Pinging 192.168.103.2 with 32 bytes of data:
Reply from 192.168.103.2: bytes=32 time<10ms TTL=128
Reply from 192.168.103.2: bytes=32 time<10ms TTL=128
Reply from 192.168.103.2: bytes=32 time<10ms TTL=128
Reply from 192.168.103.2: bytes=32 time<10ms TTL=128
```

```
Ping statistics for 192.168.103.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum =  0ms, Average =  0ms
C:\WINNT\system32>
```

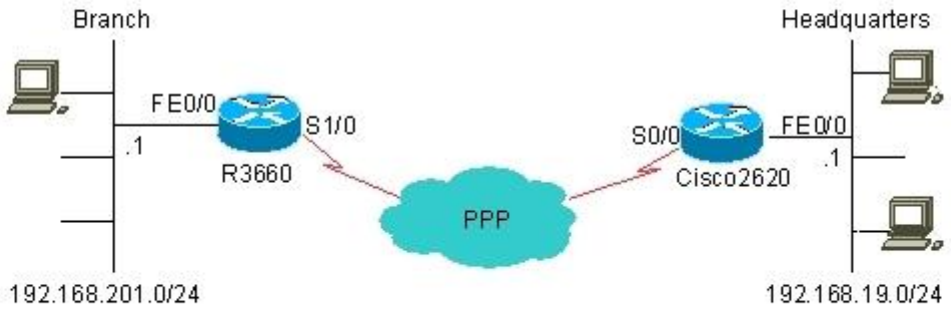
The preceding shows that the host "DENGL-NECBOOK" of the branch successfully accesses the Internet and the intranet of the headquarters. This host does not need VPDN configuration. Network administrators need to only allocate an intranet address (192.168.201.78 here) to it and set its gateway address to 192.168.201.1, which can be seen in the following information in the NAT recording node of the R3660.

```
Ruijie# show ip nat translations
Pro Inside global      Inside local        Outside local       Outside global
icmp 192.168.103.6:512  192.168.201.78:512 192.168.207.2:512  192.168.207.2:512
icmp 192.168.12.217:512 192.168.201.78:512 192.168.12.1:512   192.168.12.1:512
Ruijie# show ip route
Codes: C - connected, S - static,  R - RIP
       O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2
Gateway of last resort is 192.168.12.1 to network 0.0.0.0
     192.168.103.0/32 is subnetted, 2 subnets
C       192.168.103.6 is directly connected, Virtual-PPP1
C       192.168.103.2 is directly connected, Virtual-PPP1
C    192.168.12.0/24 is directly connected, FastEthernet0/1
C    192.168.201.0/24 is directly connected, FastEthernet0/0
S*   0.0.0.0/0 [1/0] via 192.168.12.1, FastEthernet0/1
Ruijie#
```

Establishing a Tunnel with Cisco 2620

Figure 21 shows the networking topology of the L2TP tunnel established by using Ruijie router and Cisco 2620. Cisco 2620 acts as the L2TP network server (LNS).

Figure 21 Networking topology of the L2TP tunnel established by using Cisco 2620 (LNS)



The configurations of R3660 and Cisco 2620 are respectively described as follows:

22) R3660 configuration

```
R3660# show running-config
Building configuration...
```

```
Current configuration : 1136 bytes
!
hostname R3660
access-list 1 permit any
!
l2tp-class l2x
authentication
hostname branch
password share
!
pseudowire-class pw
encapsulation l2tpv2
protocol l2tpv2 l2x
ip local interface serial1/0
!
!
!
!
!
!
interface serial 1/0
encapsulation PPP
ip address 202.101.93.21 255.255.255.192
ip nat outside
!
interface serial 1/1
clock rate 64000
!
interface serial 1/2
clock rate 64000
!
interface serial 1/3
clock rate 64000
!
interface FastEthernet 0/0
ip address 192.168.201.1 255.255.255.0
ip nat inside
duplex auto
speed auto
!
interface FastEthernet 0/1
duplex auto
speed auto
!
interface Null 0
```

```
!
interface Virtual-ppp 1
pseudowire 202.101.93.23 7 pw-class pw
ppp pap sent-username rgnos password 7 072C04211A01
ip mtu 1460
ip address 192.168.103.3 255.255.255.0
!
router ospf
network 192.168.103.0 0.0.0.255 area 0.0.0.1
network 192.168.201.0 0.0.0.255 area 0.0.0.1
!
ip nat inside source list 1 interface Serial1/0 overload
ip route 0.0.0.0 0.0.0.0 serial 1/0
ip route 192.168.19.0 255.255.255.0 Virtual-ppp 1
!
line con 0
line aux 0
line vty 0 4
!
!
end
R3660#
```

In this configuration, the intranet of the branch shares (through route setting) the L2TP tunnel that the interface virtual-ppp 1 establishes with the headquarters, to access the intranet of the headquarters. The intranet of the branch shares (using the NAT function) the WAN interface Serial 0 to access the Internet. Distribution of such data streams is controlled by means of route setting. Users can also use the interface FastEthernet 0 as the WAN interface as required (for example, using the ADSL line as the WAN line), share this interface through NAT, and use the line connected to the interface Serial 0 for connecting to the headquarters. This configuration can be seen in the routing table and the ARP table.

```
R3660# show ip route
Codes: C - connected, S - static,  R – RIP
      O - OSPF, IA - OSPF inter area
      E1 - OSPF external type 1, E2 - OSPF external type 2
Gateway of last resort is 0.0.0.0 to network 0.0.0.0
    192.168.103.0/32 is subnetted, 2 subnets
C      192.168.103.3 is directly connected, Virtual-PPP1
C      192.168.103.2 is directly connected, Virtual-PPP1
S    192.168.19.0/24 is directly connected, Virtual-PPP1
C    192.168.201.0/24 is directly connected, FastEthernet0/0
    202.101.93.0/24 is variably subnetted, 2 subnets, 2 masks
C      202.101.93.23/32 is directly connected, Serial1/0
C      202.101.93.0/26 is directly connected, Serial1/0
S*   0.0.0.0/0 is directly connected, Serial1/0
R3660#show arp
Protocol  Address        Age (min)  Hardware Addr   Type   Interface
```

```
Internet  192.168.201.213        3   0010.8801.a5c3  ARPA   FastEthernet0/0
Internet  192.168.201.1          -   00d0.f8fb.126e  ARPA   FastEthernet0/0
R3660#
```

23) Cisco 2620 configuration

```
Cisco2620# show running-config
Building configuration...
Current configuration : 1212 bytes
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Cisco2620
!
!
username 163 password 0 163
username rgnos password 0 rgnos
username 263 password 0 263
ip subnet-zero
!
!
no ip domain-lookup
!
vpdn enable
!
vpdn-group 1
! Default L2TP VPDN group
accept-dialin
protocol l2tp
virtual-template 1
l2tp tunnel password 7 0832444F1B1C
!
call rsvp-sync
!
interface FastEthernet0/0
ip address 192.168.19.1 255.255.255.0
duplex auto
speed 10
!
interface Serial0/0
ip address 202.101.93.23 255.255.255.192
encapsulation ppp
fair-queue
clockrate 2000000
```

```
!
interface Serial0/1
no ip address
shutdown
!
interface Virtual-Template1
ip address 192.168.103.2 255.255.255.0
ppp authentication pap
!
router ospf 100
log-adjacency-changes
network 192.168.103.0 0.0.0.255 area 1
!
ip classless
ip http server
!
!
voice-port 1/0/0
!
voice-port 1/0/1
!
dial-peer cor custom
!
!
gatekeeper
shutdown
!
!
line con 0
exec-timeout 0 0
line aux 0
privilege level 15
line vty 0 4
privilege level 15
no login
line vty 5 15
login
!
end
```

Different from the Windows 2000 server, L2TP on Cisco 2620 is not bound with IPSec/IKE. Nevertheless, Cisco L2TP requires tunnel authentication by default, while L2TP on the Windows 2000 server does not support tunnel authentication. Cisco 2620 learns routes reachable to the network 192.168.201.0/24 over OSPF, a dynamic routing protocol, which can be seen in the routing table.

```
Cisco2620# show ip route
```

```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B – BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2, E – EGP
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default, U - per-user static route, o - ODR
P - periodic downloaded static route
Gateway of last resort is not set
O    192.168.201.0/24 [110/20] via 192.168.103.3, 00:06:26, Virtual-Access1
192.168.103.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.103.3/32 is directly connected, Virtual-Access1
C       192.168.103.0/24 is directly connected, Virtual-Access1
202.101.93.0/24 is variably subnetted, 2 subnets, 2 masks
C       202.101.93.21/32 is directly connected, Serial0/0
C       202.101.93.0/26 is directly connected, Serial0/0
C    192.168.19.0/24 is directly connected, FastEthernet0/0
Cisco2620#
```

The following shows how the host "BLIZZARD" of the branch accesses the Internet and the intranet of the headquarters.

```
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.
C:\>ipconfig /all
Windows 2000 IP Configuration
       Host Name . . . . . . . . . . . . : BLIZZARD
       Primary DNS Suffix  . . . . . . . :
       Node Type . . . . . . . . . . . . : Broadcast
       IP Routing Enabled. . . . . . . . : No
       WINS Proxy Enabled. . . . . . . . : No
Ethernet adapter local connection 2:
       Connection-specific DNS Suffix  . :
       Description . . . . . . . . . . . : NE2000 Compatible
       Physical Address. . . . . . . . . : 00-10-88-01-A5-C3
       DHCP Enabled. . . . . . . . . . . : No
       IP Address. . . . . . . . . . . . : 192.168.201.213
       Subnet Mask . . . . . . . . . . . : 255.255.255.0
       Default Gateway . . . . . . . . . : 192.168.201.1
       DNS Servers . . . . . . . . . . . : 202.101.143.141
C:\>route print
===========================================================================
Interface List
0x1 ........................... MS TCP Loopback interface
0x3000003 ...00 10 88 01 a5 c3 ...... Novell 2000 Adapter.
===========================================================================
===========================================================================
Active Routes:
```

```
Network Destination        Netmask          Gateway       Interface  Metric
         0.0.0.0         0.0.0.0     192.168.201.1  192.168.201.213    1
       127.0.0.0       255.0.0.0       127.0.0.1       127.0.0.1    1
   192.168.201.0   255.255.255.0  192.168.201.213  192.168.201.213    1
  192.168.201.213  255.255.255.255       127.0.0.1       127.0.0.1    1
  192.168.201.255  255.255.255.255  192.168.201.213  192.168.201.213    1
       224.0.0.0       224.0.0.0  192.168.201.213  192.168.201.213    1
  255.255.255.255  255.255.255.255  192.168.201.213  192.168.201.213    1
Default Gateway:   192.168.201.1
====================================================================
Persistent Routes:
  None
C:\>ping 192.168.103.2
Pinging 192.168.103.2 with 32 bytes of data:
Reply from 192.168.103.2: bytes=32 time=10ms TTL=0
Reply from 192.168.103.2: bytes=32 time<10ms TTL=0
Reply from 192.168.103.2: bytes=32 time<10ms TTL=0
Reply from 192.168.103.2: bytes=32 time<10ms TTL=0
Ping statistics for 192.168.103.2:
   Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
   Minimum = 0ms, Maximum =  10ms, Average =  2ms
C:\>ping 192.168.19.1
Pinging 192.168.19.1 with 32 bytes of data:
Reply from 192.168.19.1: bytes=32 time<10ms TTL=254
Reply from 192.168.19.1: bytes=32 time<10ms TTL=254
Reply from 192.168.19.1: bytes=32 time<10ms TTL=254
Reply from 192.168.19.1: bytes=32 time<10ms TTL=254
Ping statistics for 192.168.19.1:
   Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
   Minimum = 0ms, Maximum =  0ms, Average =  0ms
C:\>ping 202.101.93.23
Pinging 202.101.93.23 with 32 bytes of data:
Reply from 202.101.93.23: bytes=32 time<10ms TTL=254
Reply from 202.101.93.23: bytes=32 time<10ms TTL=254
Reply from 202.101.93.23: bytes=32 time<10ms TTL=254
Reply from 202.101.93.23: bytes=32 time<10ms TTL=254
Ping statistics for 202.101.93.23:
   Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
   Minimum = 0ms, Maximum =  0ms, Average =  0ms
C:\>
```

The preceding shows that the host "BLIZZARD" of the branch successfully accesses the Internet and the intranet of the headquarters. This host does not need VPDN configuration. Network administrators need to only allocate an intranet address (192.168.201.213 here) to it and set its gateway address to 192.168.201.1, which can be seen in the following information in the NAT recording node of the R3660.

```
R3660# show ip nat translations
Pro Inside global     Inside local      Outside local      Outside global
icmp 202.101.93.21:1024 192.168.201.213:1024 202.101.93.23:1024 202.101.93.23:1024
R3660#
```

Establishing a Tunnel with the Windows 2000 PC by Using hostname

Connect Ruijie router to a Windows 2000 PC. For information about the networking topology and PC configuration, see "Establishing a Tunnel with the Windows 2000 Server."

The hostname of the Windows 2000 PC must have been registered on the DNS server. The DNS client function must be enabled on Ruijie router and route configuration must be correct so that Ruijie router can ping the DNS server successfully.

Ruijie LAC router configuration:

```
l2tp-class 1
!
pseudowire-class 1
 encapsulation l2tpv2
!
no service password-encryption
!
ip name-server 192.168.5.119
ip name-server 61.154.22.41
!
no ip ref load-sharing original
!
interface FastEthernet 0/0
 ip ref
 ip address 192.168.52.90 255.255.255.0
 duplex auto
 speed auto
!
interface FastEthernet 0/1
 duplex auto
 speed auto
!
interface Virtual-ppp 1
 pseudowire hostname mm.hxs.meibu.com 1 encapsulation l2tpv2
 ppp pap sent-username user1 password  11
 ip address negotiate
!
ip route 0.0.0.0 0.0.0.0 192.168.52.1
!
```

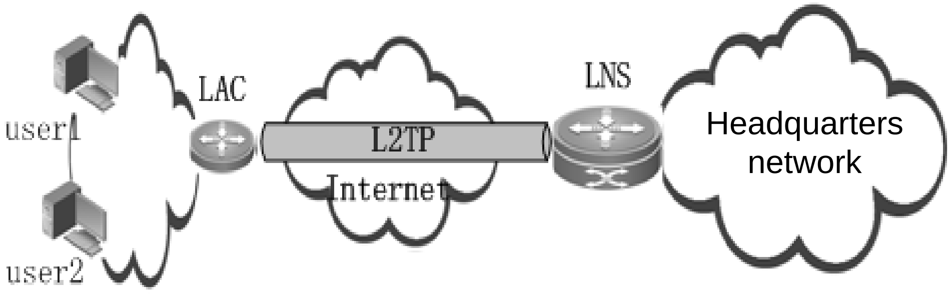
```
ref parameter 75 400
line con 0
line aux 0
line vty 0 4
 login
```

The tunnel is automatically triggered for connection.

IPv6 over L2TP Configuration Example

In the following figure, Ruijie RG-RSR20 (LAC) and RG-RSR30 (LNS) are used to establish a network connection of the L2TP tunnel. To establish an IPv6 over L2TP tunnel, the LNS assigns an IPv6 prefix to the client connected to the virtual PPP interface of the LAC over Dynamic Host Configuration Protocol version 6 Prefix Delegation (DHCPv6-PD), and delivers the domain name and DNS address over DHCPv6.

Figure 22 Establishing the IPv6 over L2TP Tunnel by RG-RSR20 (LAC) and RG-RSR30 (LNS)



The configuration of RG-RSR20 is as follows:

```
RSR20# show running-config
Building configuration...
Current configuration : 1136 bytes
!
hostname RSR20
access-list 1 permit any
!
!
l2tp-class l2x
authentication
hostname branch
password share
!
pseudowire-class pw
encapsulation l2tpv2
protocol l2tpv2 l2x
ip local interface FastEthernet 0/0
!
!
interface FastEthernet 0/0
```

```
ip address 192.168.201.1 255.255.255.0
duplex auto
speed auto
!
interface FastEthernet 0/1
duplex auto
speed auto
!
interface Null 0
!
interface Virtual-ppp 1
pseudowire 192.168.202.1 7 pw-class pw
ppp pap sent-username rgnos password 7 123
ip mtu 1460
ipv6 address prefix-from-provider ::1111/64
ipv6 enable
ipv6 dhcp client pd prefix-from-provider
!
!
line con 0
line aux 0
line vty 0 4
!
!
end
RSR20#
```

The configuration of RG-RSR30 is as follows:

```
RSR30# show running-config
Building configuration...
Current configuration : 766 bytes
!
hostname RSR30
!
ipv6 dhcp pool ipv6_dhcp_pool1
prefix-delegation pool client-prefix-pool lifetime 2000 1000
 dns-server 2001:DB8:3000:3000::42
 domain-name example.com
!
ipv6 local pool client-prefix-pool1 2001:DB8:1200::/40 48
!
vpdn enable
!
vpdn-group 1
! Default L2TP VPDN group
```

```
accept-dialin
protocol l2tp
virtual-template 1
l2tp tunnel authentication
l2tp tunnel password share
!
!
!
username rgnos password 7 123
!
interface FastEthernet 0/0
ip address 192.168.202.1 255.255.255.0
duplex auto
speed auto
!
interface FastEthernet 0/1
ip address 192.168.12.217 255.255.255.0
duplex auto
speed auto
!
interface Loopback 1
ipv6 address 2001:1200::2/64
!
interface Null 0
!
interface Virtual-Template 1
ppp authentication pap
ipv6 enable
no ipv6 nd suppress-ra
ipv6 nd managed-config-flag
ipv6 nd other-config-flag
ipv6 dhcp server ipv6_dhcp_pool1
!
!
line con 0
line aux 0
line vty 0 4
!
!
end
RSR30#
```

Initiation by the Remote Client

Configuration Task List

- Configuring a local address pool (optional)
- Configuring user information (optional)
- Setting VPDN global parameters (mandatory)
- Configuring a virtual-template interface (mandatory)
- Configuring VPDN-group (mandatory)

Configuring a Local Address Pool

This is an optional step for establishing an L2TP tunnel initiated by the remote client. In order to accept the L2TP connection initiated by the remote client, the LNS must allocate an IP address to the remote client if an IP address used inside the VPN is not set for the remote client. Generally, an idle IP address in a specified address pool is allocated to the client.

Use the following commands to configure a local address pool.

Command	Function
Ruijie(config)# **ip local pool** *poolname first-ip* [*last-ip*]	Creates or sets a local address pool.
Ruijie(config)# **no ip local pool** *poolname*	Deletes an address pool of the specified name.

poolname is the name of the local address pool to be created or set, *first-ip* is the first address in the address range set for the local address pool, and *last-ip* is the last address in the address range set for the local address pool.

Configuring User Information

This is an optional step for establishing an L2TP tunnel initiated by the remote client. The purpose of configuring user information is to authenticate remote L2TP clients that attempt to access the local client.

Use the following commands to configure user information.

Command	Function
Ruijie(config)# **username** *user-name* **password {0|7}** *password*	Configures user information.
Ruijie(config)# **no username** *user-name*	Deletes the specified user.

user-name is the name of the dial-in user who is allowed to access, and *password* is the password of the user. The router locally maintains a database that records names of dial-in users permitted to access and their passwords.

Setting VPDN Global Parameters

This is a mandatory step for establishing an L2TP tunnel initiated by the remote client. VPDN global parameters are set in this step. The operations of setting VPDN global parameters include:

- Enabling or disabling the VPDN function
- Setting the VPDN source address

- Setting the maximum number of VPDN sessions
- Setting the domain resolution option
- Enabling or disabling domain authentication
- Ignoring the source-address check of VPDN
- Setting VPDN rate limit
- Enabling or disabling the VPDN function is mandatory and setting the VPDN source address is optional.

Enabling or Disabling the VPDN Function

If a user requires the router to accept the L2TP access from the remote client and establish an L2TP tunnel and session, the VPDN function must be enabled on the router.

Use the following commands to enable or disable the VPDN function.

Command	Function
Ruijie(config)# **vpdn enable**	Enables the VPDN function.
Ruijie(config)# **no vpdn enable**	Disables the VPDN function.

The VPDN enabling and disabling support instant configuration and use (that is, they are available immediately when they are configured). If the VPDN function is disabled, all the existing L2TP tunnels and sessions will be disconnected.

Setting the VPDN Source Address

After the VPDN source address is set, the destination address of the tunnel set for the remote client must be consistent with the VPDN source address before an L2TP tunnel is established successfully.

Use the following commands to set the VPDN (local) source address.

Command	Function
Ruijie(config)# **vpdn source-ip** *ip-address*	Sets the VPDN source address.
Ruijie(config)# **no vpdn source-ip** *ip-address*	Cancels the set VPDN source address.

The system does not check whether the destination address in the received tunnel establishment request is a specific value by default.

Setting the Maximum Number of VPDN Sessions

After the maximum number of VPDN sessions is set, access requests beyond the maximum value will be denied.

Use the following commands to set the maximum number of sessions supported by the VPDN server.

Command	Function
Ruijie(config)# **vpdn session-limit** *sessions*	Sets the maximum number of VPDN sessions.
Ruijie(config)# **no vpdn session-limit**	Restores the default maximum number of VPDN sessions.

The maximum number of sessions is the one configured by using this command by default.

Setting the Domain Resolution Option

Use the following commands to set the domain resolution option in VPDN domain authentication. Domains of different types can be identified based on configuration.

Command	**Function**
Ruijie(config)**# vpdn domain-delimiter @/%#-\**	Sets the domain delimiter, prefix or suffix.
Ruijie(config)**# no vpdn domain-delimiter**	Cancels the VPDN domain authentication option.

The system does not resolve the domain field by default.

Enabling or Disabling Domain Authentication

Use the following commands to enable or disable the VPDN domain authentication, that is, determine whether to strip the domain field.

Command	**Function**
Ruijie(config)**# vpdn authorize domain split**	Enables domain authentication, that is, enables domain splitting.
Ruijie(config)**# no vpdn authorize domain**	Disables domain authentication.

Domain authentication is disabled by default.

Ignoring the Source Address Check of VPDN

Use the following commands to ingore errors on received L2TP control packets that do not comply with the RFC specifications so as to ensure the normal negotiation.

Command	**Function**
Ruijie(config-vpdn)# **vpdn ignore_source**	Ignores the source address check of packets sent from the peer end.
Ruijie(config-vpdn)# **no vpdn ignore_source**	Strictly checks the source address of packets sent from the peer end.

The system strictly checks the source address by default.

Setting VPDN Rate Limit

Use the following commands to limit the rate of establishing VPDN tunnels, namely, to limit the number of VPDN tunnels that can be established at one time.

Command	**Function**
Ruijie(config)# **vpdn limit_rate** *rate_num*	Enables rate limit. *rate_num* is the number of tunnels that can be established at a time. The value range is 5 to 100.
Ruijie(config)# **no vpdn limit_rate**	Disables rate limit.

Rate limit is disabled by default.

Configuring Rate Limit Function for VPDN Transmission

Ruijie products enable users to limit the L2TP session creation rate on the LAC, so as to limit the number of ICQ packets that can be sent at a time. To configure the rate limit function, run the following commands:

Command	Function
Ruijie(config)# **vpdn send limit_rate** *rate_num*	Enables the rate limit function. *rate_num*: Indicates the number of ICRQ packets that can be sent at a time. The value range is from **4** to **100**.
Ruijie(config)# **no send vpdn limit_rate**	Disables the rate limit function.

The rate limit function is disabled by default.

Configuring a Virtual-Template Interface

This is a mandatory step for establishing an L2TP tunnel initiated by the remote client. This interface will become the template of the virtual-access interface that binds and carries L2TP sessions. The operations of configuring a virtual-template interface include:

For information about setting the local IP address, setting the authentication mode, and setting the peer IP address, see sections regarding the interface configuration guide.

Setting a Virtual-Template Interface

Use the following commands to set a virtual-template interface.

Command	Function
Ruijie(config)**# interface virtual-template** *number*	Creates or configures a specified virtual-template interface.
Ruijie(config)**# no interface virtual-template** *number*	Deletes the specified virtual-template interface.

number is the sequence number of the specified virtual-template interface. The created virtual-template interface will be used as the configuration template of the virtual-access interface that binds and carries L2TP sessions.

Configuring VPDN-group

This is a mandatory step for establishing an L2TP tunnel initiated by the remote client. VPDN-group parameters are set in this step. The operations of configuring VPDN-group include:

- Setting a VPDN-group interface (mandatory)
- Setting the tunneling mode (mandatory)
- Setting the tunneling protocol (mandatory)
- Setting a virtual template to be used (mandatory)
- Setting the peer name (optional)
- Setting the local name (optional)
- Setting the VPDN-group source address (optional)
- Setting the L2TP control connection parameters (optional)
- Setting L2TP data transmission parameters (optional)

- Setting the vrf option (optional)
- Setting the supported domain name (optional)
- Re-performing PPP negotiation (optional)
- Ignoring errors on control packets (optional)

Setting a VPDN-group Interface

Use the following commands to set a VPDN-group interface.

Command	Function
Ruijie(config)# **vpdn-group** *name*	Creates or configures the specified VPDN-group interface.
Ruijie(config)# **no vpdn-group** *name*	Deletes the specified VPDN-group interface.

name is the name of the specified VPDN-group interface. The created VPDN-group interface allows related clients to access and establish tunnels.

Setting the Tunneling Mode

Use the following commands to set the tunneling mode.

Command	Function
Ruijie (config-vpdn)# **accept-dialin**	Allows access from the dial-in remote client.
Ruijie (config-vpdn)# **no accept-dialin**	Denies the access from the dial-in remote client.

If a user needs the local router to provide the LNS function, access from the dial-in remote client must be allowed.

Setting the Tunneling Protocol

Use the following commands to set the tunneling protocol.

Command	Function
Ruijie(config-vpdn-acc-in)# **protocol** {**any** | **l2tp** | **pptp**}	Sets the tunneling protocol.
Ruijie(config-vpdn-acc-in)# **no protocol**	Cancels the set tunneling protocol.

The tunneling mode must be set before a tunneling protocol is set.

Setting a Virtual Template to Be Used

Use the following commands to set a virtual template used by the VPDN-group.

Command	Function
Ruijie(config-vpdn-acc-in)# **virtual-template** *number*	Sets a virtual template to be used.
Ruijie(config-vpdn-acc-in)# **no virtual-template**	Cancels the virtual template.

The tunneling mode must be set before a virtual template is set for the VPDN-group.

Setting the Peer Name

If the peer name is set, this VPDN-group is effective only to the remote client that matches the host name. If no peer name is set, this VPDN-group is the default VPDN-group and can provide the VPDN service for any remote client.

Use the following commands to set the peer name.

Command	Function
Ruijie(config-vpdn)# **terminate-from hostname** *name*	Sets the name of the peer host.
Ruijie(config-vpdn)# **no terminate-from**	Cancels the set peer name.

name is the name of the peer host.

Setting the Local Name

The local name is sent to the peer end as a record property.

Use the following commands to set the local name.

Command	Function
Ruijie(config-vpdn)# **local name** *name*	Sets the local name.
Ruijie(config-vpdn)# **no local name**	Cancels the set local name.

name is the local name. The router name is used as the local name and sent to the peer host of the tunnel by default.

Setting the VPDN-group Source Address

The destination address in the tunnel establishment request sent from the remote client must match the VPDN-group source adress. In this way, the VPDN-group can be applied.

Use the following commands to set the VPDN-group source address.

Command	Function
Ruijie(config-vpdn)# **source-ip** *src-ip*	Sets the VPDN-group source address.
Ruijie(config-vpdn)# **no source-ip**	Cancels the set VPDN-group source address.

src-ip is the VPDN-group source address.

Setting the L2TP Control Connection

Use the following commands to set the L2TP control connection.

Command	Function
Ruijie(config-vpdn)# **l2tp tunnel authentication**	Enables tunnel authentication.
Ruijie(config-vpdn)# **no l2tp tunnel authentication**	Disables tunnel authentication.
Ruijie(config-vpdn)# **l2tp tunnel hello** *interval*	Sets the interval of sending Hello messages.
Ruijie(config-vpdn)# **no l2tp tunnel hello**	Deletes the set interval of sending Hello messages.

Command	Function
Ruijie(config-vpdn)# **l2tp tunnel password** *pass-word*	Sets the tunnel password.
Ruijie(config-vpdn)# **no l2tp tunnel password**	Deletes the set tunnel password.
Ruijie(config-vpdn)# **l2tp tunnel receive-window** *size*	Sets the size of the receiving window of the tunnel control connection.
Ruijie(config-vpdn)# **no l2tp tunnel receive- window**	Restores the default size of the receiving window of the tunnel control connection.
Ruijie(config-vpdn)# **l2tp tunnel retransmit** {**retries** *number* | **timeout** {**min** | **max**} *seconds*}	Sets retransmission of tunnel control messages.
Ruijie(config-vpdn)# **no l2tp tunnel retransmit** {**retries** | **timeout** {**min**|**max**}}	Restores the default retransmission setting of tunnel control messages.
Ruijie(config-vpdn)# **l2tp tunnel timeout** {**no-session**| **setup**} *seconds*	Sets the maximum interval of establishing a no-session/control connection of the tunnel.
Ruijie(config-vpdn)# **no l2tp tunnel timeout** {**no-session** | **setup**}	Restores the default maximum interval of establishing a no-session/control connection of the tunnel.
Ruijie(config-vpdn)# **l2tp tunnel force_ipsec**	Enables forced encryption. It is used when external encryption is required. After this command is executed, packets can be sent to VPDN tunnels only after encryption.
Ruijie(config-vpdn)# **no l2tp tunnel force_ipsec**	Disables forced encryption.
Ruijie(config-vpdn)# **l2tp tunnel avp-hidden-compatible**	Supports the RFC2661 standard AVP Hidden parsing algorithm compatibly.
Ruijie(config-vpdn)# **no l2tp tunnel avp-hidden-compatible**	Restores the default Cisco standard AVP Hidden parsing algorithm.

Tunnel authentication is not needed for establishing L2TP tunnels by default (but required on Cisco devices by default). The default interval of sending Hello control messages is 60 seconds. The default receiving window size of control messages is 4. The default number of retransmission times of control messages is 5. The default minimum and maximum intervals of retransmitting control messages are 1 second and 8 seconds respectively. The default maximum interval of no session in a tunnel is 600 seconds. The default maximum time supported by tunnels in establishing a control connection is 300 seconds. If L2TP tunnel authentication is required, the same tunnel password must be configured at both ends of the L2TP tunnel. Nevertheless, the system does not configure tunnel passwords for any L2TP tunnels by default and does not require tunnel authentication. The *interval* parameter is the interval of sending Hello messages, in seconds. The unit of the *seconds* parameter is also seconds. The forced IPSec encryption and authentication are disabled by default. The default AVP Hidden parsing algorithm uses the Cisco standard. After the RFC2661 standard AVP Hidden parsing algorithm is supported compatibly, the RFC2661 standard is used to parse and hide AVP.

Setting L2TP Data Transmission Parameters

Use the following commands to set IP/UDP parameters for transmitting L2TP messages.

Command	Function
Ruijie(config-vpdn)# **l2tp ip udp checksum**	Sets the UDP checksum.

Ruijie(config-vpdn)# **no l2tp ip udp Checksum**	Cancels the UDP checksum setting.
Ruijie(config-vpdn)# **ip tos** *tos-value*	Sets the IP TOS field.
Ruijie(config-vpdn)# **no ip tos**	Cancels the IP TOS setting.
Ruijie(config-vpdn)# **ip precedence** *value*	Sets the IP Precedence field.
Ruijie(config-vpdn)# **no ip precedence**	Cancels the IP Precedence setting.

tos-value is the value of the TOS field of the IP header that carries L2TP messages, and *value* is the value of the Precedence field of this IP header. For L2TP messages to be carried in L2TP tunnels of RGOS, the checksum field of UDP that carries L2TP messages must be blank, the TOS of the IP header that carries L2TP messages and the Precedence field of this IP header must be 0 by default.

Note that the TOS and Precedence fields are supported only in L2TP. Though they can be configured in PPTP, the configuration does not take effect.

Setting the VRF Option

Use the following commands to set the VRF to which specified L2TP tunnel packets belong. The configuration maps to the **ip vrf forward** command of the VT interface, implementing VRF spanning.

Command	Function
Ruijie(config-vpdn)#**vpn vrf** *vrf-name*	Sets the VRF attribute for the tunnel.
Ruijie(config-vpdn)# **no vpn vrf**	Deletes the VRF attribute setting of the tunnel.

vrf-name is the name of the VRF.

Setting the Supported Domain Name

The command for setting the supported domain name takes effect after the domain authentication is enabled. Only the domain matching the content of this command can be identified. If a domain does not match the content of this command, another VPDN group is used for matching. If no matched VPDN group is found, the authentication fails.

Use the following commands to set the domain name.

Command	Function
Ruijie(config-vpdn)# **domain** *domain-name* **vrf** *vrf-name*	Sets the authentication domain name and the related VRF.
Ruijie(config-vpdn)# **no domain** *domain-name*	Cancels the domain setting.

domain-name is the name of domain, and *vrf-name* is the name of the VRF.

Setting the Domain Name-Bound Address Pool

The command for setting the domain name-based address pool is configured in **vpdn-domain** command mode. Domain names are authenticated and information about the bound address pool configured using this command is obtained during VPDN tunnel negotiation. After the PPP negotiation is successful, the specified bound address pool is used to assign peer addresses of tunnels. The address pool configured in the virtual-template interface is used for address assignment by default.

Use the following commands to set the domain name-bound address pool.

Command	**Function**
Ruijie(config-vpdn)# **domain** *domain-name* **vrf** *vrf-name* Ruijie(config-vpdn-domain)# **pool** *pool-name*	Sets the address pool bound to the authentication domain name.
Ruijie(config-vpdn)# **no domain** *domain-name*	Cancels the address pool binding.

domain-name is the name of the domain, *vrf-name* is the name of the VRF, and *pool-name* is the name of the address pool.

Setting the DNS Negotiation Address of PPP Bound to a Domain Name

The command for setting the DNS negotiation address of PPP bound to a domain name is configured in **vpdn-domain** command mode. The domain name is authenticated and information about the bound DNS negotiation address of PPP configured using this command is obtained during VPDN tunnel negotiation. This address is used for DNS negotiation during PPP negotiation. The DNS address of PPP configured in the virtual-template interface is used for negotiation by default.

Use the following commands to set addresses used by DNS during PPP negotation by matching domain names.

Command	**Function**
Ruijie(config-vpdn)# **domain** *domain-name* **vrf** *vrf-name* Ruijie(config-vpdn-domain)#**dns** *A.B.C.D A.B.C.D*	Sets the DNS negotiation address of PPP bound to the authentication domain name.
Ruijie(config-vpdn)# **no dns**	Cancels the DNS binding.

domain-name is the name of the domain, *vrf-name* is the name of VRF, and *A.B.C.D* is the DNS address.

Re-Performing PPP Authentication

When the client triggers the LAC to start dialing, the LAC acts as the LNS to authenticate the client. This command is used to perform CHAP authentication on the client again after an L2TP tunnel is established. This commad is valid only on the LNS.

Use the following commands to forcibly perform complete PPP authentication again.

Command	**Function**
Ruijie(config-vpdn)# **force-local-chap**	Forces the LNS to perform CHAP authentication on the client again.
Ruijie(config-vpdn)# **no force-local-chap**	Cancels CHAP re-authentication.

Re-Performing PPP Negotiation

When the client triggers the LAC to start dialing, the LAC acts as the LNS to negotiate with the client. This command is used to perform LCP negotiation with the client again after an L2TP tunnel is established. This commad is valid only on the LNS.

Use the following commands to forcibly perform PPP negotiation again.

Command	**Function**

Ruijie(config-vpdn)# **force-local-lcp**	Forces the LNS to perform LCP negotiation with the client again.
Ruijie(config-vpdn)# **no force-local-lcp**	Cancels LCP re-negotiation.

Ignoring Errors on Control Packets

Use the following commands to ignore errors on received L2TP control packets that do not comply with the RFC specifications to ensure normal negotiation.

Command	Function
Ruijie(config-vpdn)# **lcp renegotiation always**	Ignores errors on packets from the peer end.
Ruijie(config-vpdn)# **no lcp renegotiation always**	Checks whether control packets comply with RFC specifications.

Configuring the Function of Not Carrying Tunnel Authentication Response AVP in SCCRP Packets

This command enables the L2TP LNS to return SCCRP packets without Challenge Response AVP 13 when receiving SCCRQ packets without Challenge AVP 11 after tunnel authentication is configured. You can run the no form of this command to disable this function. This command is mainly used to ensure compatibility with the SCCRQ link detection function of devices provided by some manufacturer. It avoids returning SCCRP packets with all-zero authentication information when the other party sends SCCRQ packets without authentication information.

Command	Function
Ruijie(config-vpdn)#**l2tp tunnel zxkeepalive-compatible**	Returns SCCRP packets without the challenge response AVP upon receiving SCCRQ packets without the challenge AVP.
Ruijie(config-vpdn)#**no l2tp tunnel zxkeepalive-compatible**	Returns SCCRP packets with the all-zero challenge response AVP upon receiving SCCRQ packets without the challenge AVP.

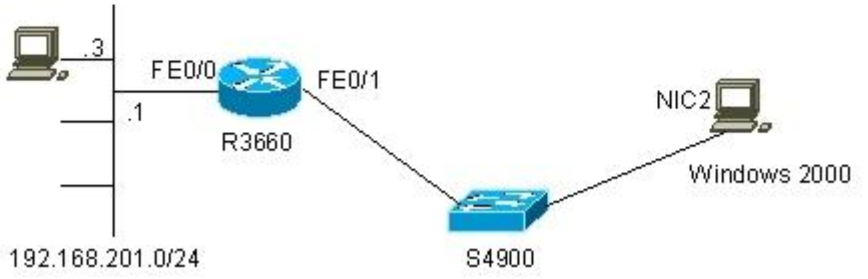
Configuration Examples

Three configuration examples are given below. In one configuration example, the Windows 2000 server is used as the remote L2TP client for access, and tunnel authentication is not performed due to limitations of the Windows 2000 server. In another example, Cisco 3640 is used as the remote L2TP client, and tunnel authentication is required. In the other example, Cisco 2620 is used as the L2TP client, and tunnel authentication is required.

Establishing a Tunnel with the Windows 2000 Server

Figure shows the networking topology of the L2TP tunnel established by using Ruijie router R3660 and the Windows 2000 server.

Figure 23 Networking topology of the L2TP tunnel established by using the Windows 2000 server (LAC)



The configurations of the R3660 and Windows 2000 server are respectively described as follows:

24) R3660 configuration:

```
R3660# show running-config
Building configuration...
Current configuration : 708 bytes
!
hostname R3660
!
vpdn enable
!
vpdn-group 1
! Default L2TP VPDN group
accept-dialin
protocol l2tp
virtual-template 1
!
!
!
username rgnos password 7 04251F083110
!
ip local pool vpdnusers 192.168.101.3 192.168.101.253
!
interface FastEthernet 0/0
ip address 192.168.201.1 255.255.255.0
duplex auto
speed auto
!
interface FastEthernet 0/1
ip address 192.168.12.217 255.255.255.0
duplex auto
speed auto
!
interface Loopback 1
ip address 192.168.101.2 255.255.255.0
!
interface Null 0
!
```

```
interface Virtual-Template 1
ppp authentication pap
ip unnumbered Loopback 1
peer default ip address pool vpdnusers
!
!
line con 0
line aux 0
line vty 0 4
!
!
end
R3660#
```

25) Configuration of the Windows 2000 server:

```
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.
C:\>ipconfig /all
Windows 2000 IP Configuration
       Host Name . . . . . . . . . . . . : BLIZZARD
       Primary DNS Suffix  . . . . . . . :
       Node Type . . . . . . . . . . . . : Broadcast
       IP Routing Enabled. . . . . . . . : No
       WINS Proxy Enabled. . . . . . . . : No
Ethernet adapter local connection 2:
       Connection-specific DNS Suffix  . :
       Description . . . . . . . . . . . : NE2000 Compatible
       Physical Address. . . . . . . . . : 00-10-88-01-A5-C3
       DHCP Enabled. . . . . . . . . . . : No
       IP Address. . . . . . . . . . . . : 192.168.12.213
       Subnet Mask . . . . . . . . . . . : 255.255.255.0
       Default Gateway . . . . . . . . . : 192.168.12.1
       DNS Servers . . . . . . . . . . . : 202.101.143.141
C:\>route print
==========================================================================
Interface List
0x1 ........................... MS TCP Loopback interface
0x2000003 ...00 10 88 01 a5 c3 ...... Novell 2000 Adapter.
==========================================================================
==========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0     192.168.12.1  192.168.12.213    1
        127.0.0.0        255.0.0.0        127.0.0.1       127.0.0.1    1
     192.168.12.0    255.255.255.0  192.168.12.213  192.168.12.213    1
```

```
   192.168.12.213  255.255.255.255       127.0.0.1       127.0.0.1     1
   192.168.12.255  255.255.255.255   192.168.12.213  192.168.12.213     1
        224.0.0.0        224.0.0.0   192.168.12.213  192.168.12.213     1
  255.255.255.255  255.255.255.255   192.168.12.213  192.168.12.213     1
Default Gateway:     192.168.12.1
===========================================================================
Persistent Routes:
  None
C:\>
```

In **Network and Dial-up Connections**, click **New connection** and select **Connect to a private network through the Internet** to establish a virtual private connection to the specified LNS, namely, R3660. Set the destination address to **192.168.12.217**. In the properties, set **Security measure** to **Advanced (user-defined setting)**. In **Advanced security settings**, set **Data encryption** to **Optional encryption (connect even if no encryption available)**, click **Password not encrypted (PAP)** to use PAP as the authentication protocol, and click **OK** to save these property settings. Then, you can use the user **RGNOS** set the R3660 and its password to establish a virtual connection.

The following shows the routing and communication information after a virtual connection is established successfully.

26) R3660:

```
R3660# show ip route
Codes: C - connected, S - static,  R - RIP
O - OSPF, IA - OSPF inter area
E1 - OSPF external type 1, E2 - OSPF external type 2
Gateway of last resort is not set
192.168.101.0/24 is variably subnetted, 2 subnets, 2 masks
C      192.168.101.8/32 is directly connected, Virtual-Access1
C      192.168.101.0/24 is directly connected, Loopback1
C    192.168.12.0/24 is directly connected, FastEthernet0/1
C    192.168.201.0/24 is directly connected, FastEthernet0/0
R3660#ping 192.168.101.8
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echoes to 192.168.101.8, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
R3660#show vpdn tunnel
L2TP Tunnel Information Total tunnels 1
LocID RemID Remote Name   State  Remote Address  Port  Sessions L2TP Class/
                                                     VPDN Group
7    8    BLIZZARD     est    192.168.12.213  1701  1     1
%No active PPTP tunnels
R3660#
R3660# show vpdn session
L2TP Session Information Total sessions 1
LocID     RemID     TunID     Username, Intf/     State
Last Chg
```

```
                           Vcid, Circuit
1       1        7        ,Vi1           est    00:02:08
%No active PPTP tunnels
R3660#
```

27) Windows 2000 server:

```
Microsoft Windows 2000 [Version 5.00.2195]
(C) Copyright 1985-2000 Microsoft Corp.
C:\>ipconfig /all
Windows 2000 IP Configuration
       Host Name . . . . . . . . . . . . : BLIZZARD
       Primary DNS Suffix  . . . . . . . :
       Node Type . . . . . . . . . . . . : Broadcast
       IP Routing Enabled. . . . . . . . : No
       WINS Proxy Enabled. . . . . . . . : No
Ethernet adapter local connection 2:
       Connection-specific DNS Suffix  . :
       Description . . . . . . . . . . . : NE2000 Compatible
       Physical Address. . . . . . . . . : 00-10-88-01-A5-C3
       DHCP Enabled. . . . . . . . . . . : No
       IP Address. . . . . . . . . . . . : 192.168.12.213
       Subnet Mask . . . . . . . . . . . : 255.255.255.0
       Default Gateway . . . . . . . . . : 192.168.12.1
       DNS Servers . . . . . . . . . . . : 202.101.143.141
PPP adapter L2TP:
       Connection-specific DNS Suffix  . :
       Description . . . . . . . . . . . : WAN (PPP/SLIP) Interface
       Physical Address. . . . . . . . . : 00-53-45-00-00-00
       DHCP Enabled. . . . . . . . . . . : No
       IP Address. . . . . . . . . . . . : 192.168.101.8
       Subnet Mask . . . . . . . . . . . : 255.255.255.255
       Default Gateway . . . . . . . . . : 192.168.101.8
       DNS Servers . . . . . . . . . . . :
C:\>route print
==========================================================================
Interface List
0x1 ........................... MS TCP Loopback interface
0x2000003 ...00 10 88 01 a5 c3 ...... Novell 2000 Adapter.
0xd000004 ...00 53 45 00 00 00 ...... WAN (PPP/SLIP) Interface
==========================================================================
==========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
         0.0.0.0          0.0.0.0     192.168.12.1  192.168.12.213      2
         0.0.0.0          0.0.0.0    192.168.101.8   192.168.101.8      1
       127.0.0.0        255.0.0.0        127.0.0.1       127.0.0.1      1
```

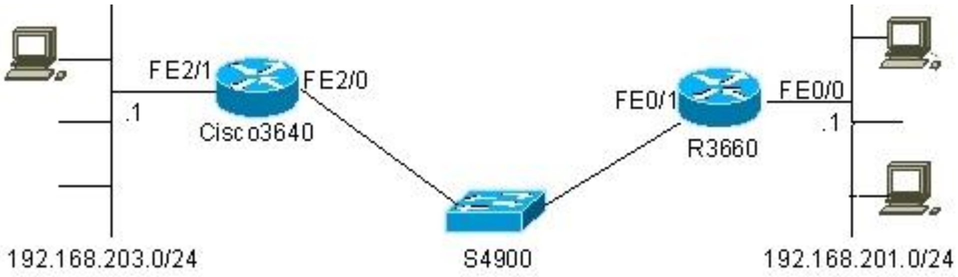
```
    192.168.12.0    255.255.255.0  192.168.12.213  192.168.12.213     1
   192.168.12.213 255.255.255.255        127.0.0.1      127.0.0.1     1
   192.168.12.217 255.255.255.255   192.168.12.213 192.168.12.213     1
   192.168.12.255 255.255.255.255   192.168.12.213 192.168.12.213     1
   192.168.101.2  255.255.255.255   192.168.101.8    192.168.101.8    1
   192.168.101.8  255.255.255.255        127.0.0.1      127.0.0.1     1
  192.168.101.255 255.255.255.255    192.168.101.8   192.168.101.8    1
        224.0.0.0       224.0.0.0   192.168.12.213  192.168.12.213    1
        224.0.0.0       224.0.0.0    192.168.101.8    192.168.101.8   1
  255.255.255.255 255.255.255.255   192.168.12.213  192.168.12.213    1
Default Gateway:   192.168.101.8
==========================================================================
Persistent Routes:
  None
C:\>ping 192.168.101.2
Pinging 192.168.101.2 with 32 bytes of data:
Reply from 192.168.101.2: bytes=32 time<10ms TTL=255
Reply from 192.168.101.2: bytes=32 time<10ms TTL=255
Reply from 192.168.101.2: bytes=32 time<10ms TTL=255
Reply from 192.168.101.2: bytes=32 time<10ms TTL=255
Ping statistics for 192.168.101.2:
   Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
   Minimum = 0ms, Maximum =  0ms, Average =  0ms
C:\>ping 192.168.201.1
Pinging 192.168.201.1 with 32 bytes of data:
Reply from 192.168.201.1: bytes=32 time<10ms TTL=255
Reply from 192.168.201.1: bytes=32 time<10ms TTL=255
Reply from 192.168.201.1: bytes=32 time<10ms TTL=255
Reply from 192.168.201.1: bytes=32 time<10ms TTL=255
Ping statistics for 192.168.201.1:
   Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
   Minimum = 0ms, Maximum =  0ms, Average =  0ms
C:\>ping 192.168.201.3
Pinging 192.168.201.3 with 32 bytes of data:
Reply from 192.168.201.3: bytes=32 time<10ms TTL=254
Reply from 192.168.201.3: bytes=32 time<10ms TTL=254
Reply from 192.168.201.3: bytes=32 time<10ms TTL=254
Reply from 192.168.201.3: bytes=32 time<10ms TTL=254
Ping statistics for 192.168.201.3:
   Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
   Minimum = 0ms, Maximum =  0ms, Average =  0ms
C:\>ping 192.168.12.1
```

```
Pinging 192.168.12.1 with 32 bytes of data:
Reply from 192.168.12.1: bytes=32 time=10ms TTL=255
Reply from 192.168.12.1: bytes=32 time<10ms TTL=255
Reply from 192.168.12.1: bytes=32 time<10ms TTL=255
Reply from 192.168.12.1: bytes=32 time<10ms TTL=255
Ping statistics for 192.168.12.1:
   Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
   Minimum = 0ms, Maximum =  10ms, Average =  2ms
C:\>
```

Establishing a Tunnel with Cisco 3640

Figure   shows the networking topology of the L2TP tunnel established by using Ruijie router R3660 and Cisco 3640.

Figure 24 Networking topology of the L2TP tunnel established with Cisco 3640 (LAC)



The configurations of R3660 and Cisco 3640 are respectively described as follows:

28) R3660 tconfiguration:

```
R3660# show running-config
Building configuration...
Current configuration : 766 bytes
!
hostname R3660
!
vpdn enable
!
vpdn-group 1
! Default L2TP VPDN group
accept-dialin
protocol l2tp
virtual-template 1
l2tp tunnel authentication
l2tp tunnel password share
!
!
!
username rgnos password 7 025144391715
!
```

```
ip local pool vpdnusers 192.168.101.3 192.168.101.253
!
interface FastEthernet 0/0
ip address 192.168.201.1 255.255.255.0
duplex auto
speed auto
!
interface FastEthernet 0/1
ip address 192.168.12.217 255.255.255.0
duplex auto
speed auto
!
interface Loopback 1
ip address 192.168.101.2 255.255.255.0
!
interface Null 0
!
interface Virtual-Template 1
ppp authentication pap
ip unnumbered Loopback 1
peer default ip address pool vpdnusers
!
!
line con 0
line aux 0
line vty 0 4
!
!
end
R3660#
```

29) Cisco 3640 configuration:

```
C3640# show running-config
Building configuration...
Current configuration : 2096 bytes
!
version 12.3
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
no service dhcp
!
hostname C3640
!
!
ip subnet-zero
```

```
!
l2tp-class l2x
authentication
password 0 share
!
pseudowire-class pw
encapsulation l2tpv2
protocol l2tpv2 l2x
ip local interface FastEthernet2/0
!
no mpls ldp logging neighbor-changes
no scripting tcl init
no scripting tcl encdir
!
no voice hpi capture buffer
no voice hpi capture destination
!
controller E1 3/0
channel-group 1 timeslots 1-2
!
!
!
interface Loopback0
ip address 132.11.10.2 255.255.255.0
!
interface FastEthernet2/0
ip address 192.168.12.242 255.255.255.0
speed auto
duplex auto
!
interface FastEthernet2/1
ip address 192.168.203.1 255.255.255.0
duplex auto
speed auto
!
interface Serial3/0:1
ip address 192.168.1.2 255.255.255.0
!
interface Virtual-PPP1
ip address negotiated
no cdp enable
ppp pap sent-username rgos password 0 rgos
pseudowire 192.168.12.217 11 pw-class pw
!
no ip http server
```

```
ip classless
ip route 0.0.0.0 0.0.0.0 FastEthernet2/0 192.168.12.1
!
!
dial-peer cor custom
!
dial-peer voice 111 voip
session protocol sipv2
codec g711alaw
!
!
line con 0
exec-timeout 0 0
line aux 0
line vty 0 4
privilege level 15
no login
line vty 5 871
login
!
!
end
```

The following shows the routing and communication information on the R3660 and Cisco 3640 after an L2TP tunnel is established.

30) R3660:

```
R3660# show ip route
Codes: C - connected, S - static,  R - RIP
O - OSPF, IA - OSPF inter area
E1 - OSPF external type 1, E2 - OSPF external type 2
Gateway of last resort is not set
192.168.101.0/24 is variably subnetted, 2 subnets, 2 masks
C      192.168.101.9/32 is directly connected, Virtual-Access1
C      192.168.101.0/24 is directly connected, Loopback1
C   192.168.12.0/24 is directly connected, FastEthernet0/1
C   192.168.201.0/24 is directly connected, FastEthernet0/0
R3660# show vpdn
L2TP Tunnel and Session Information Total tunnels 1 sessions 1
LocID RemID Remote Name   State  Remote Address  Port  Sessions L2TP Class/
                                                     VPDN Group
9    21511 C3640        est    192.168.12.242  1701  1       1
LocID     RemID     TunID     Username, Intf/     State  Last Chg
                           Vcid, Circuit
1      14        9         ,Vi1           est    00:18:06
%No active PPTP tunnels
```

```
R3660#ping 192.168.101.9
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echoes to 192.168.101.9, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
R3660#
```

The preceding information shows that an L2TP tunnel has been established successfully and communication can be made successfully.

31) Cisco 3640:

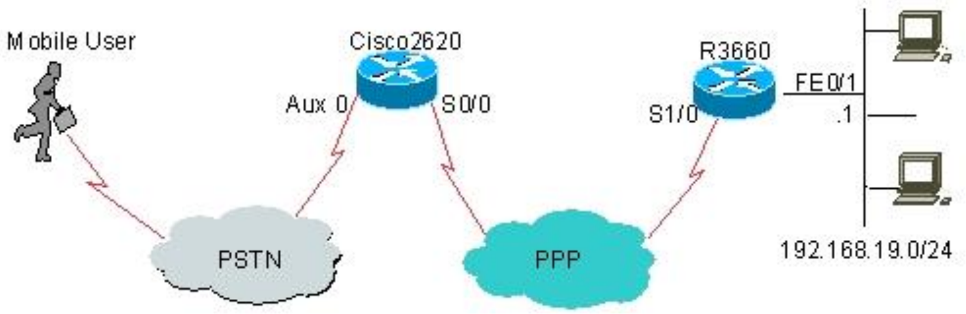
```
C3640# show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route
Gateway of last resort is 192.168.12.1 to network 0.0.0.0
C    192.168.12.0/24 is directly connected, FastEthernet2/0
     132.11.0.0/24 is subnetted, 1 subnets
C    132.11.10.0 is directly connected, Loopback0
C    192.168.1.0/24 is directly connected, Serial3/0:1
     192.168.101.0/32 is subnetted, 2 subnets
C    192.168.101.9 is directly connected, Virtual-PPP1
C    192.168.101.2 is directly connected, Virtual-PPP1
S*   0.0.0.0/0 [1/0] via 192.168.12.1, FastEthernet2/0
C3640# show vpdn
%No active L2F tunnels
L2TP Tunnel and Session Information Total tunnels 1 sessions 1
LocID RemID Remote Name   State  Remote Address  Port  Sessions L2TP Class/
                                                       VPDN Group
21511 9    Ruijie        est    192.168.12.217  1701  1       l2x
LocID     RemID     TunID     Username, Intf/     State  Last Chg Uniq ID
                              Vcid, Circuit
14        1         21511      11, Vp1            est    00:23:58 2
%No active PPTP tunnels
C3640# ping 192.168.101.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echoes to 192.168.101.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
C3640#
```

The preceding information shows that an L2TP tunnel has been established successfully on Cisco 3640 and communication can be made successfully.

Establishing a Tunnel with Cisco 2620

Figure shows the networking topology of the L2TP tunnel established by using the Ruijie router R3660 and Cisco 2620. Here, Cisco 2620 actually works as both the access server (AS) and LAC. The AS and LAC functions are generally provided by ISPs, and devices with the same functions but with more powerful performance, such as Cisco AS5300 or Cisco AS5800, are often used to provide the functions. Cisco 2620 is used to establish an L2TP tunnel here. The L2TP tunnel is established between Cisco 2620 and the R3660, but PPP negotiation is performed between mobile or dial-up users (mobile user in Figure) and the R3660. The mobile user needs to only configure a dial-up connection and connect to the server in dial-up mode by using the allocated user name and password, which are not closely related to L2TP here. This is also an advantage of this tunneling mode. L2TP tunnel settings are transparent to users. The mobile user configuration and usage are not described here. For information about dial-up setting, see instructions of the related operating system.

Figure 25 Networking topology of the L2TP tunnel established with Cisco 2620 (LAC)



The configurations of R3660 and Cisco 2620 are respectively described as follows:

32) R3660 configuration:

```
R3660# show running-config
Building configuration...
Current configuration : 989 bytes
!
hostname R3660
!
vpdn enable
!
vpdn-group 1
! Default L2TP VPDN group
accept-dialin
protocol l2tp
virtual-template 1
l2tp tunnel authentication
l2tp tunnel password share
!
!
!
username rgnos password 7 025144391715
username pc@i-net.com.cn password 7 127654431B
!
```

```
ip local pool vpdnusers 192.168.101.3 192.168.101.253
!
interface serial 1/0
encapsulation PPP
ip address 202.101.93.21 255.255.255.0
!
interface serial 1/1
clock rate 64000
!
interface serial 1/2
clock rate 64000
!
interface serial 1/3
clock rate 64000
!
interface FastEthernet 0/0
duplex auto
speed auto
!
interface FastEthernet 0/1
ip address 192.168.19.1 255.255.255.0
duplex auto
speed auto
!
interface Loopback 1
ip address 192.168.101.2 255.255.255.0
!
interface Null 0
!
interface Virtual-Template 1
ppp authentication pap
ip unnumbered Loopback 1
peer default ip address pool vpdnusers
!
!
line con 0
line aux 0
line vty 0 4
!
!
end
R3660#
```

33) Cisco 2620 configuration:

```
Cisco2620# show running-config
Building configuration...
```

```
Current configuration : 1677 bytes
!
version 12.3
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
no service dhcp
!
hostname Cisco2620
!
!
username pc password 0 1111
username 163 password 0 163
no aaa new-model
ip subnet-zero
!
!
!
vpdn enable
!
vpdn-group 1
request-dialin
protocol l2tp
domain i-net.com.cn
initiate-to ip 202.101.93.21
l2tp tunnel password 7 0832444F1B1C
!
interface FastEthernet0/0
ip address 192.168.7.1 255.255.255.0
duplex auto
speed 10
!
interface Serial0/0
ip address 202.101.93.23 255.255.255.0
encapsulation ppp
fair-queue
clockrate 2000000
!
interface Serial0/1
no ip address
shutdown
!
interface Async65
ip address 5.5.5.5 255.255.255.0
encapsulation ppp
```

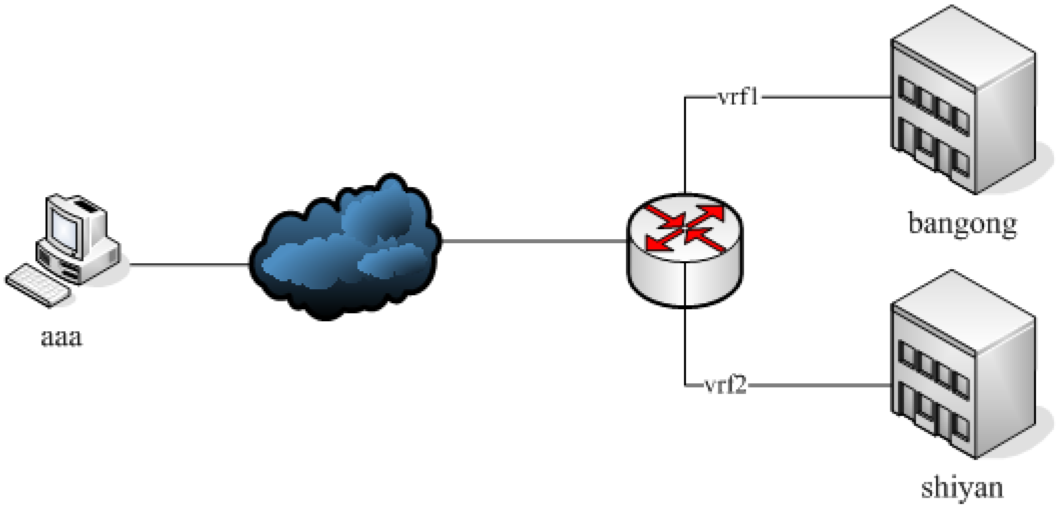
```
dialer in-band
dialer idle-timeout 30000
dialer string 8435
dialer-group 1
async mode dedicated
peer default ip address 5.5.5.6
ppp authentication pap
!
no ip http server
ip classless
!
dialer-list 1 protocol ip permit
!
!
line con 0
exec-timeout 0 0
line aux 0
login local
modem InOut
transport input all
autoselect during-login
autoselect ppp
line vty 0 4
privilege level 15
no login
line vty 5 15
login
!
no scheduler allocate
end
Cisco2620#
```

The mobile user can access the intranet 192.168.19.0/24 connected to the R3660 through the L2TP tunnel simply by using the user name **pc@i-net.com.cn** and password **1111**.

Configuration Example of Domain Authentication

The following figure shows the networking topology for one user to connect to two networks by using one public network address based on domain information.

Figure 26



```
ip vrf vrf1
ip vrf vrf2
!
vpdn enable
vpdn domain-delimiter @/%#-\ suffix
vpdn authorize domain split
!
vpdn-group 1
! Default L2TP VPDN group
accept-dialin
protocol l2tp
virtual-template 1
domain bangong vrf vrf1   /*Specify the domain name*/
domain shiyan vrf vrf2   /*Specify the domain name*/
!
!
username rgos password 7 025144391715
username pc@i-net.com.cn password 7 127654431B
!
ip local pool vpdnusers 192.168.101.3 192.168.101.253
!
interface FastEthernet 0/1
ip address 192.168.19.1 255.255.255.0
duplex auto
speed auto
!
interface Loopback 1
ip vrf forward vrf1
ip address 192.168.101.2 255.255.255.0
!
interface Loopback 2
ip vrf forward vrf2
ip address 192.168.101.2 255.255.255.0
!
interface Null 0
!
interface Virtual-Template 1
ppp authentication pap
ip unnumbered Loopback 1
peer default ip address pool vpdnusers
!
!
line con 0
line aux 0
line vty 0 4
!
!
End
```

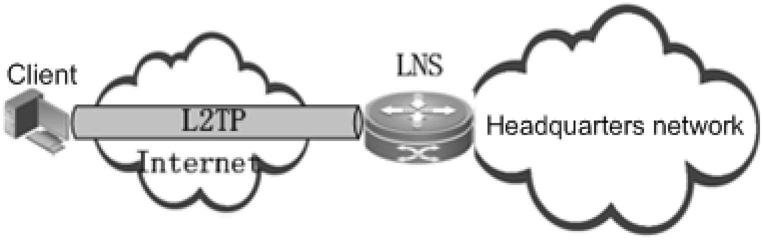
Note

When you need to connect to the same LNS in dial-up mode on the same device, VP ports must belong to different VRFs if different domains are used. Otherwise, the same destination address will be assigned to two VP ports, the routes of the two VP ports are dupliated, and only one route is available. On the LNS side, two VA ports belong to different VRFs, and packets forwarded by different tunnels will be sent to different VRFs. Packets sent from the LNS to one VP port may be returned by the other VP port, resulting in route dissymmetry. In this case, data forwarding is not affected, but pinging the peer address from the LNS may fail.

IPv6 over L2TP

In the following figure, Ruijie RG-RSR30 (LNS) and a client establish the IPv6 over L2TP tunnel. The LNS assigns the IPv6 address to the client via ND, and delivers the domain name and DNS address over DHCPv6.

Figure 27 Establishing IPv6 over L2TP Tunnel by RG-RSR30 (LNS) and Client



The configuration of RG-RSR30 is as follows:

```
RSR30# show running-config
Building configuration...
Current configuration : 766 bytes
```

```
!
hostname RSR30
!
ipv6 dhcp pool ipv6_dhcp_pool1
 dns-server 2001:DB8:3000:3000::42
 domain-name example.com
iana-address prefix 2001:1200::/64 lifetime 2000 1000
!
vpdn enable
!
vpdn-group 1
! Default L2TP VPDN group
accept-dialin
protocol l2tp
virtual-template 1
!
!
!
username rgnos password 7 025144391715
!
!
interface FastEthernet 0/0
ip address 192.168.201.1 255.255.255.0
duplex auto
speed auto
!
interface FastEthernet 0/1
ip address 192.168.12.217 255.255.255.0
duplex auto
speed auto
!
interface Loopback 1
ipv6 address 2001:1200::2/64
!
interface Null 0
!
interface Virtual-Template 1
ppp authentication pap
ipv6 enable
no ipv6 nd suppress-ra
ipv6 nd managed-config-flag
ipv6 nd other-config-flag
ipv6 unnumbered Loopback 1
ipv6 dhcp server ipv6_dhcp_pool1
!
```

```
!
line con 0
line aux 0
line vty 0 4
!
!
end
RSR30#
```

Monitoring and Maintaining L2TP Tunnels

RGOS provides L2TP monitoring and maintenance functions.

Monitoring L2TP Tunnels

Use the following commands to monitor L2TP tunnels.

Command	Function
Ruijie# **show vpdn** [**session | tunnel** [**l2tp** *locid*]] Or Ruijie> **show vpdn** [**session | tunnel** [**l2tp** *locid*]]	Displays information about the current VPDN tunnel and session. Displays information about the tunnel of the specified ID.
Ruijie# **debug vpdn error**	Enables the VPDN error debugging function.
Ruijie# **no debug vpdn error**	Disables the VPDN error debugging function.
Ruijie# **debug vpdn event**	Enables the VPDN event debugging function.
Ruijie# **no debug vpdn event**	Disables the VPDN event debugging function.
Ruijie# **debug vpdn packet**	Enables the VPDN packet debugging function.
Ruijie# **no debug vpdn packet**	Disables the VPDN packet debugging function.
Ruijie# **debug vpdn l2x-data**	Enables the VPDN l2x-data debugging function.
Ruijie# **no debug vpdn l2x-data**	Disables the VPDN l2x-data debugging function.
Ruijie# **debug vpdn l2x-errors**	Enables the VPDN l2x-errors debugging function.
Ruijie# **no debug vpdn l2x-errors**	Disables the VPDN l2x-errors debugging function.
Ruijie# **debug vpdn l2x-events**	Enables the VPDN l2x-events debugging function.
Ruijie# **no debug vpdn l2x-events**	Disables the VPDNl2x-events debugging function.
Ruijie# **debug vpdn l2x-packets**	Enables the VPDN l2x-packets debugging function.
Ruijie# **no debug vpdn l2x-packets**	Disables the VPDN l2x-packets debugging function.

Displaying Information About the Current L2TP Tunnel

You can use the **show vpdn** command in real time as required to view information about the current L2TP tunnel (including channel information and session information).

Note

The length of usernames is unlimited. The **show vpdn** command displays only the first 12 charaters of a username for the sake of format alignment. You can use the **show vpdn tunnel l2tp** *locid* command to view the full username.

```
Ruijie# show vpdn tunnel
L2TP Tunnel Information Total tunnels 1
LocID RemID Remote Name   State  Remote Address  Port  Sessions L2TP Class/
                                                     VPDN Group
1    35390 C3640       est   192.168.12.242  1701  1       1
%No active PPTP tunnels
Ruijie# show vpdn session
L2TP Session Information Total sessions 1
LocID     RemID     TunID     Username, Intf/     State
Last Chg
                           Vcid, Circuit
1        1261      1         rgnos,Vi1              est
01:04:42
%No active PPTP tunnels
Ruijie# show vpdn
L2TP Tunnel and Session Information Total tunnels 1 sessions 1
LocID RemID Remote Name   State  Remote Address  Port  Sessions L2TP Class/
                                                     VPDN Group
1    35390 C3640       est   192.168.12.242  1701  1       1
LocID     RemID     TunID     Username, Intf/     State  Last Chg
                           Vcid, Circuit
1        1261      1         rgnos,Vi1              est
01:04:45
%No active PPTP tunnels
Ruijie#
Ruijie# show vpdn tunnel l2tp 1
L2TP tunnel locid 1 is up, remote id is 35390, 1 active sessions
  Tunnel state is est
  Tunnel transport is UDP
  Remote tunnel name is C3640
    Internet Address 192.168.12.242, port 1701
  Local tunnel name is Ruijie
    Internet Address 192.168.12. 217, port 1701
  VPDN group for tunnel is 1
  Tunnel domain unknown
  ip mtu adjust disabled
  Control Ns 2, Nr 4
```

Performing Overall VPDN Debugging

RGOS provides VPDN debugging functions, which are useful to both L2TP and PPTP. The following is an overall VPDN debugging example, in which the LNS accepts the dial-in request from the peer end and finally establishes a tunnel (including channels and sessions).

```
Ruijie# debug vpdn error
vpdn protocol errors debugging is on
Ruijie# debug vpdn event
vpdn events debugging is on
Ruijie# debug vpdn packet
vpdn packet debugging is on
Ruijie# show debug
VPDN:
vpdn events debugging is on
vpdn protocol errors debugging is on
vpdn packet debugging is on
Ruijie#
VPDN PROCESS From tunnel: Received 158 byte pak
L2X: UDP socket write 168 bytes, 192.168.12.217(1701) to 192.168.12.242(1701)
L2X: UDP socket write 40 bytes, 192.168.12.217(1701) to 192.168.12.242(1701)
VPDN PROCESS From tunnel: Pak consumed
VPDN PROCESS From tunnel: Received 70 byte pak
L2X: UDP socket write 40 bytes, 192.168.12.217(1701) to 192.168.12.242(1701)
VPDN PROCESS From tunnel: Pak consumed
VPDN PROCESS From tunnel: Received 76 byte pak
Get virtual-access from free queue: Virtual-Access1
Clone virtual-access from interface Virtual-Template1
L2X: UDP socket write 56 bytes, 192.168.12.217(1701) to 192.168.12.242(1701)
L2X: UDP socket write 40 bytes, 192.168.12.217(1701) to 192.168.12.242(1701)
VPDN PROCESS From tunnel: Pak consumed
VPDN PROCESS From tunnel: Received 76 byte pak
L2X: UDP socket write 40 bytes, 192.168.12.217(1701) to 192.168.12.242(1701)
Vi1 Tnl/Sn 3/1 L2TP: Virtual interface created for unknown, bandwidth 1024 Kbps
Vi1 Tnl/Sn 3/1 L2TP: VPDN session up
VPDN PROCESS From tunnel: Pak consumed
VPDN PROCESS From tunnel: Received 50 byte pak
Vi1 VPDN PROCESS From tunnel: Queue 14 byte pak to ppp parse and iqueue
Vi1 VPDN PROCESS From tunnel: Pak send successful
%UPDOWN: Interface Virtual-Access1, changed state to up
Vi1 VPDN PROCESS Into tunnel: Sending 54 byte pak
L2X: UDP socket write 54 bytes, 255.255.255.255(1701) to 4.83.68.68(1701)
VPDN PROCESS From tunnel: Received 50 byte pak
Vi1 VPDN PROCESS From tunnel: Queue 14 byte pak to ppp parse and iqueue
Vi1 VPDN PROCESS From tunnel: Pak send successful
Vi1 VPDN PROCESS Into tunnel: Sending 50 byte pak
```

```
L2X: UDP socket write 50 bytes, 255.255.255.255(1701) to 4.83.68.68(1701)
Vi1 VPDN PROCESS Into tunnel: Sending 54 byte pak
L2X: UDP socket write 54 bytes, 255.255.255.255(1701) to 4.83.68.68(1701)
VPDN PROCESS From tunnel: Received 50 byte pak
Vi1 VPDN PROCESS From tunnel: Queue 14 byte pak to ppp parse and iqueue
Vi1 VPDN PROCESS From tunnel: Pak send successful
Vi1 VPDN PROCESS Into tunnel: Sending 50 byte pak
L2X: UDP socket write 50 bytes, 255.255.255.255(1701) to 4.83.68.68(1701)
Vi1 VPDN PROCESS Into tunnel: Sending 54 byte pak
L2X: UDP socket write 54 bytes, 255.255.255.255(1701) to 4.83.68.68(1701)
VPDN PROCESS From tunnel: Received 50 byte pak
Vi1 VPDN PROCESS From tunnel: Queue 14 byte pak to ppp parse and iqueue
Vi1 VPDN PROCESS From tunnel: Pak send successful
Vi1 VPDN PROCESS Into tunnel: Sending 50 byte pak
L2X: UDP socket write 50 bytes, 192.168.12.217(1701) to 192.168.12.242(1701)
Vi1 VPDN PROCESS Into tunnel: Sending 54 byte pak
L2X: UDP socket write 54 bytes, 192.168.12.217(1701) to 192.168.12.242(1701)
VPDN PROCESS From tunnel: Received 54 byte pak
Vi1 VPDN PROCESS From tunnel: Queue 18 byte pak to ppp parse and iqueue
Vi1 VPDN PROCESS From tunnel: Pak send successful
VPDN PROCESS From tunnel: Received 56 byte pak
Vi1 VPDN PROCESS From tunnel: Queue 20 byte pak to ppp parse and iqueue
Vi1 VPDN PROCESS From tunnel: Pak send successful
Vi1 VPDN PROCESS Into tunnel: Sending 45 byte pak
L2X: UDP socket write 45 bytes, 192.168.12.217(1701) to 192.168.12.242(1701)
Vi1 VPDN PROCESS Into tunnel: Sending 50 byte pak
L2X: UDP socket write 50 bytes, 192.168.12.217(1701) to 192.168.12.242(1701)
VPDN PROCESS From tunnel: Received 50 byte pak
Vi1 VPDN PROCESS From tunnel: Queue 14 byte pak to ppp parse and iqueue
Vi1 VPDN PROCESS From tunnel: Pak send successful
Vi1 VPDN PROCESS Into tunnel: Sending 50 byte pak
L2X: UDP socket write 50 bytes, 192.168.12.217(1701) to 192.168.12.242(1701)
VPDN PROCESS From tunnel: Received 50 byte pak
Vi1 VPDN PROCESS From tunnel: Queue 14 byte pak to ppp parse and iqueue
Vi1 VPDN PROCESS From tunnel: Pak send successful
VPDN PROCESS From tunnel: Received 50 byte pak
Vi1 VPDN PROCESS From tunnel: Queue 14 byte pak to ppp parse and iqueue
Vi1 VPDN PROCESS From tunnel: Pak send successful
Vi1 VPDN PROCESS Into tunnel: Sending 50 byte pak
L2X: UDP socket write 50 bytes, 192.168.12.217(1701) to 192.168.12.242(1701)
%UPDOWN: Line protocol on Interface Virtual-Access1, changed state to up
Ruijie# show ip route
Codes: C - connected, S - static,  R - RIP
O - OSPF, IA - OSPF inter area
E1 - OSPF external type 1, E2 - OSPF external type 2
```

```
Gateway of last resort is not set
192.168.101.0/24 is variably subnetted, 2 subnets, 2 masks
C      192.168.101.5/32 is directly connected, Virtual-Access1
C      192.168.101.0/24 is directly connected, Virtual-Access1
C   192.168.12.0/24 is directly connected, FastEthernet0
C   192.168.201.0/24 is directly connected, Ethernet0
Ruijie#
```

Performing L2TP Data Debugging

If a user needs to check whether L2TP can send control messages successfully, enable the L2TP data debugging function. The following is an L2TP data debugging example, in which the LNS accepts the dial-in request from the peer end and finally establishes a tunnel (including channels and sessions) after the l2x-data debugging function is enabled.

```
Ruijie# no debug all
All possible debugging has been turned off
Ruijie# debug vpdn l2x-data
L2X data packets debugging is on
Ruijie#
L2X: Punting to L2TP control message queue
L2X: Punting to L2TP control message queue
L2X: Punting to L2TP control message queue
L2X: Punting to L2TP control message queue
L2X: Punting to L2TP control message queue
L2X: Punting to L2TP control message queue
%UPDOWN: Interface Virtual-Access1, changed state to up
%UPDOWN: Line protocol on Interface Virtual-Access1, changed state to up
Ruijie#
```

Performing L2TP Error Debugging

Users can enable the l2x-erorrs debugging function to check whether a tunnel estabishment failure results from configuration inconsistency at both ends (for example, different tunnel authentication passwords at both ends). The following is an example of errors reported due to a tunnel authentcation failure.

```
Ruijie# no debug all
All possible debugging has been turned off
Ruijie# debug vpdn l2x-errors
L2X protocol errors debugging is on
Ruijie# config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# interface virtual-ppp 1
Ruijie(config-if)# no shutdown
Ruijie(config-if)# end
Ruijie#
Tnl 14 L2TP: Tunnel auth failed for BLIZZARD
Tnl 14 L2TP: Expected
```

```
9E 8D 7A 8E 78 EA 41 9F A1 74 01 21 DE 4F F3 F0
Tnl 14 L2TP: Got
84 E5 62 69 AE 46 A5 98 4E FE E2 38 EE F2 B7 E2
Ruijie# no debug all
All possible debugging has been turned off
Ruijie#
```

Performing L2TP Event Debugging

Users can enable the l2x-events debugging function to check the entire process of L2TP tunnel negotiation and establishment. The following is an L2TP event debugging example, in which the LNS accepts the dial-in request from the peer end and finally establishes a tunnel (including channels and sessions) after the l2x-events debugging function is enabled. The tunnel authentication function is enabled here.

```
Ruijie# show vpdn tunnel
%No active L2TP tunnels
%No active PPTP tunnels
Ruijie# no debug all
All possible debugging has been turned off
Ruijie# debug vpdn l2x-events
L2X protocol events debugging is on
Ruijie#
L2TP: I SCCRQ from C3640 tnl 26656
New tunnel created for remote C3640, address 192.168.12.242
Tnl 0 L2TP: Got a challenge in SCCRQ, C3640
Tnl 20 L2TP: O SCCRP  to C3640 tnlid 26656
Tnl 20 L2TP: Control channel retransmit delay set to 1 seconds
Tnl 20 L2TP: Tunnel state change from idle to wait-ctl-conn
Tnl 20 L2TP: I SCCCN from C3640 tnl 26656
Tnl 20 L2TP: Got a Challenge Response in SCCCN, C3640
Tnl 20 L2TP: Tunnel Authentication success
Tnl 20 L2TP: Tunnel state change from wait-ctl-conn to established
Tnl 20 L2TP: SM State established
Tnl 20 L2TP: I ICRQ from C3640 tnl 26656
Tnl/Sn 20/1 L2TP: Accepted ICRQ, new session created
Tnl/Sn 20/1 L2TP: O ICRP to C3640 26656/1279
Tnl/Sn 20/1 L2TP: Session state change from idle to wait-connect
Tnl 20 L2TP: Control channel retransmit delay set to 1 seconds
Tnl/Sn 20/1 L2TP: I ICCN from C3640 tnl 26656, cl 1279
Tnl/Sn 20/1 L2TP: Session state change from wait-connect to wait-for-service-sel
ection-iccn
Vi1 Tnl/Sn 20/1 L2TP: Session state change from wait-for-service-selection- iccn to established
%UPDOWN: Interface Virtual-Access1, changed state to up
%UPDOWN: Line protocol on Interface Virtual-Access1, changed state to up
Ruijie# show ip route
Codes: C - connected, S - static,  R - RIP
```

```
O - OSPF, IA - OSPF inter area
E1 - OSPF external type 1, E2 - OSPF external type 2
Gateway of last resort is not set
192.168.101.0/24 is variably subnetted, 2 subnets, 2 masks
C      192.168.101.7/32 is directly connected, Virtual-Access1
C      192.168.101.0/24 is directly connected, Virtual-Access1
C   192.168.12.0/24 is directly connected, FastEthernet0
C   192.168.201.0/24 is directly connected, Ethernet0
Ruijie# show vpdn
L2TP Tunnel and Session Information Total tunnels 1 sessions 1
LocID RemID Remote Name   State  Remote Address  Port  Sessions L2TP Class/
                                                       VPDN Group
20   26656 C3640       est   192.168.12.242  1701  1    1
LocID   RemID   TunID   Username, Intf/    State  Last Chg
                            Vcid, Circuit
1     1279      20        rgnos,Vi1      est   00:00:38
%No active PPTP tunnels
Ruijie#
```

The following is an L2TP event debugging example, in which the LNS accesses the remote L2TP server and finally establishes a tunnel (including channels and sessions) after the l2x-events debugging function is enabled. The tunnel authentication function is disabled here.

```
Ruijie# no debug all
All possible debugging has been turned off
Ruijie# debug vpdn l2x-events
L2X protocol events debugging is on
Ruijie# config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# interface virtual-ppp 1
Ruijie(config-if)# no shut
Ruijie(config-if)# end
Ruijie#
Tnl 21 L2TP: SM State idle
Tnl 21 L2TP: O SCCRQ
Tnl 21 L2TP: Control channel retransmit delay set to 1 seconds
Tnl 21 L2TP: Tunnel state change from idle to wait-ctl-reply
Tnl 21 L2TP: SM State wait-ctl-reply
Tnl 21 L2TP: O Resend SCCRQ, flg TLS, ver 2, len 96, tnl 0, ns 0, nr 0
Tnl 21 L2TP: Control channel retransmit delay set to 1 seconds
Tnl 21 L2TP: I SCCRP from
Tnl 21 L2TP: O SCCCN  to BLIZZARD tnlid 40
Tnl 21 L2TP: Control channel retransmit delay set to 1 seconds
Tnl 21 L2TP: Tunnel state change from wait-ctl-reply to established
Tnl 21 L2TP: SM State established
Vi1 Tnl/Sn 21/1 L2TP: O ICRQ to BLIZZARD 40/0
```

```
Vi1 Tnl/Sn 21/1 L2TP: Control channel retransmit delay set to 1 seconds
Vi1 Tnl/Sn 21/1 L2TP: Session state change from wait-for-tunnel to wait-reply
Vi1 Tnl/Sn 21/1 L2TP: I ICRP from BLIZZARD
Vi1 Tnl/Sn 21/1 L2TP: O ICCN to BLIZZARD 40/1
Vi1 Tnl/Sn 21/1 L2TP: Control channel retransmit delay set to 1 seconds
Vi1 Tnl/Sn 21/1 L2TP: Session state change from wait-reply to established
%UPDOWN: Interface Virtual-PPP1, changed state to up
%UPDOWN: Line protocol on Interface Virtual-PPP1, changed state to up
Ruijie# show vpdn
L2TP Tunnel and Session Information Total tunnels 1 sessions 1
LocID RemID Remote Name   State  Remote Address  Port  Sessions L2TP Class/
                                                       VPDN Group
21    40    BLIZZARD      est    192.168.12.213  1701  1
LocID    RemID   TunID    Username, Intf/    State  Last Chg
                            Vcid, Circuit
1      1        21        13,Vi1          est    00:00:27
%No active PPTP tunnels
Ruijie#
```

Performing L2TP Message Data Debugging

L2TP message data debugging refers to displaying the content of an L2TP control message after a user enables the l2x-packets debugging function. The following is an L2TP message data debugging example, in which the LNS accepts the dial-in request from the peer end and finally establishes a tunnel (including channels and sessions) after the l2x-packets debugging function is enabled. The tunnel authentication function is enabled here.

```
Ruijie# no debug all
All possible debugging has been turned off
Ruijie# debug vpdn l2x-packets
L2X control packets debugging is on
Ruijie# show vpdn
%No active L2TP tunnels
%No active PPTP tunnels
Ruijie#
L2TP: I SCCRQ from C3640 tnl 18889
L2X: Parse  AVP 0, len 8, flag 0x8000 (M)
L2X: Parse SCCRQ
L2X: Parse  AVP 2, len 8, flag 0x8000 (M)
L2X: Protocol Ver 1
L2X: Parse  AVP 6, len 8, flag 0x0
L2X: Firmware Ver 0x1130
L2X: Parse  AVP 7, len 11, flag 0x8000 (M)
L2X: Hostname C3640
L2X: Parse  AVP 8, len 25, flag 0x0
L2X: Vendor Name Cisco Systems, Inc.
L2X: Parse  AVP 10, len 8, flag 0x8000 (M)
```

```
L2X: Rx Window Size 800
L2X: Parse  AVP 11, len 22, flag 0x8000 (M)
L2X: Chlng
       98 20 4E 34 6A 4C E1 E7 FA CF 58 07 FF 4E 56 A3
L2X: Parse  AVP 9, len 8, flag 0x8000 (M)
L2X: Assigned Tunnel ID 18889
L2X: Parse  AVP 3, len 10, flag 0x8000 (M)
L2X: Framing Cap 0x3
L2X: Parse  AVP 4, len 10, flag 0x8000 (M)
L2X: Bearer Cap 0x3
L2X: No missing AVPs in SCCRQ
L2X: I SCCRQ, flg TLS, ver 2, len 130, tnl 0, ns 0, nr 0 contiguous pak, size 130
C8 02 00 82 00 00 00 00 00 00 00 00 80 08 00 00
00 00 00 01 80 08 00 00 00 02 01 00 00 08 00 00
00 06 11 30 80 0B 00 00 00 07 43 33 36 34 30 00
19 00 00 00 08 43 69 73 63 6F 20 53 79 73 74 65
6D 73 2C 20 49 6E 63 2E ...
Tnl 22 L2TP: O SCCRP  to C3640 tnlid 18889
Tnl 22 L2TP: O SCCRP, flg TLS, ver 2, len 140, tnl 18889, ns 0, nr 1
C8 02 00 8C 49 C9 00 00 00 00 00 01 80 08 00 00
00 00 00 02 80 08 00 00 00 02 01 00 80 0A 00 00
00 03 00 00 00 01 80 0A 00 00 00 04 00 00 00 00
00 08 00 00 00 06 11 30 80 0A 00 00 00 07 52 36
32 31 00 0E 00 00 00 08 ...
Tnl 22 L2TP: O ZLB ctrl ack, flg TLS, ver 2, len 12, tnl 18889, ns 1, nr 1
C8 02 00 0C 49 C9 00 00 00 01 00 01
Tnl 22 L2TP: Parse  AVP 0, len 8, flag 0x8000 (M)
Tnl 22 L2TP: Parse SCCCN
Tnl 22 L2TP: I SCCCN from C3640 tnl 18889
Tnl 22 L2TP: Parse  AVP 13, len 22, flag 0x8000 (M)
Tnl 22 L2TP: Chlng Resp
5C D5 A4 37 36 A6 7D 0F FE EF 22 48 B8 DF F5 12
Tnl 22 L2TP: No missing AVPs in SCCCN
Tnl 22 L2TP: I SCCCN, flg TLS, ver 2, len 42, tnl 22, ns 1, nr 1 contiguous pak, size 42
C8 02 00 2A 00 16 00 00 00 01 00 01 80 08 00 00
00 00 00 03 80 16 00 00 00 0D 5C D5 A4 37 36 A6
7D 0F FE EF 22 48 B8 DF F5 12
Tnl 22 L2TP: O ZLB ctrl ack, flg TLS, ver 2, len 12, tnl 18889, ns 1, nr 2
C8 02 00 0C 49 C9 00 00 00 01 00 02
Tnl 22 L2TP: Parse  AVP 0, len 8, flag 0x8000 (M)
Tnl 22 L2TP: Parse ICRQ
Tnl 22 L2TP: I ICRQ from C3640 tnl 18889
Tnl 22 L2TP: Parse  AVP 15, len 10, flag 0x8000 (M)
Tnl 22 L2TP: Serial Number -1714567290
Tnl 22 L2TP: Parse  AVP 14, len 8, flag 0x8000 (M)
Tnl 22 L2TP: Assigned Call ID 1280
Tnl 22 L2TP: Parse  AVP 18, len 10, flag 0x8000 (M)
Tnl 22 L2TP: Bearer Type 0
Tnl 22 L2TP: No missing AVPs in ICRQ
Tnl 22 L2TP: I ICRQ, flg TLS, ver 2, len 48, tnl 22, ns 2, nr 1 contiguous pak,size 48
C8 02 00 30 00 16 00 00 00 02 00 01 80 08 00 00
00 00 00 0A 80 0A 00 00 00 0F 99 CD C7 86 80 08
00 00 00 0E 05 00 80 0A 00 00 00 12 00 00 00 00
Tnl/Sn 22/1 L2TP: O ICRP to C3640 18889/1280
Tnl/Sn 22/1 L2TP: O ICRP, flg TLS, ver 2, len 28, tnl 18889, lsid 1, rsid 1280,ns 1, nr 3
C8 02 00 1C 49 C9 05 00 00 01 00 03 80 08 00 00
00 00 00 0B 80 08 00 00 00 0E 00 01
Tnl 22 L2TP: O ZLB ctrl ack, flg TLS, ver 2, len 12, tnl 18889, ns 2, nr 3
       C8 02 00 0C 49 C9 00 00 00 02 00 03
Tnl/Sn 22/1 L2TP: I ICCN from C3640 tnl 18889, cl 1280
Tnl/Sn 22/1 L2TP: Parse  AVP 0, len 8, flag 0x8000 (M)
Tnl/Sn 22/1 L2TP: Parse ICCN
Vi1 Tnl/Sn 22/1 L2TP: Parse  AVP 24, len 10, flag 0x8000 (M)
Vi1 Tnl/Sn 22/1 L2TP: Connect Speed 0
Vi1 Tnl/Sn 22/1 L2TP: Parse  AVP 19, len 10, flag 0x8000 (M)
Vi1 Tnl/Sn 22/1 L2TP: Framing Type 1
Tnl/Sn 22/1 L2TP: No missing AVPs in ICCN
Tnl/Sn 22/1 L2TP: I ICCN, flg TLS, ver 2, len 48, tnl 22, lsid 1, rsid 1280, ns 3, nr 2 contiguous
pak, size 48
C8 02 00 30 00 16 00 01 00 03 00 02 80 08 00 00
00 00 00 0C 80 0A 00 00 00 18 00 00 00 00 80 0A
00 00 00 13 00 00 00 01 00 08 00 00 00 1D 00 04
Tnl 22 L2TP: O ZLB ctrl ack, flg TLS, ver 2, len 12, tnl 18889, ns 2, nr 4
C8 02 00 0C 49 C9 00 00 00 02 00 04
%UPDOWN: Interface Virtual-Access1, changed state to up
%UPDOWN: Line protocol on Interface Virtual-Access1, changed state to up
Ruijie# show vpdn
L2TP Tunnel and Session Information Total tunnels 1 sessions 1
LocID RemID Remote Name   State  Remote Address  Port  Sessions L2TP Class/
                                                     VPDN Group
22   18889 C3640    est   192.168.12.242  1701  1       1
LocID   RemID   TunID   Username, Intf/   State  Last Chg
                           Vcid, Circuit
1      1280     22       ,Vi1           est   00:00:19
%No active PPTP tunnels
Ruijie# show ip route
Codes: C - connected, S - static,  R - RIP
O - OSPF, IA - OSPF inter area
E1 - OSPF external type 1, E2 - OSPF external type 2
Gateway of last resort is not set
```

```
192.168.101.0/24 is variably subnetted, 2 subnets, 2 masks
C      192.168.101.8/32 is directly connected, Virtual-Access1
C      192.168.101.0/24 is directly connected, Virtual-Access1
C   192.168.12.0/24 is directly connected, FastEthernet0
C   192.168.201.0/24 is directly connected, Ethernet0
Ruijie#
```

The following is an L2TP message data debugging example, in which the LAC accesses the remote L2TP server and finally establishes a tunnel (including channels and sessions), after the l2x-packets debugging function is enabled. The tunnel authentication function is disabled here.

```
Ruijie# no debug all
All possible debugging has been turned off
Ruijie# debug vpdn l2x-packets
L2X control packets debugging is on
Ruijie# config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# interface virtual-ppp 1
Ruijie(config-if)# no shutdown
Ruijie(config-if)# end
Ruijie#
Tnl 21 L2TP: O SCCRQ
Tnl 21 L2TP: O SCCRQ, flg TLS, ver 2, len 96, tnl 0, ns 0, nr 0
C8 02 00 60 00 00 00 00 00 00 00 00 80 08 00 00
00 00 00 01 80 08 00 00 00 02 01 00 80 0A 00 00
00 03 00 00 00 01 80 0A 00 00 00 04 00 00 00 00
00 08 00 00 00 06 11 30 80 0A 00 00 00 07 52 36
32 31 00 0E 00 00 00 08 ...
Tnl 21 L2TP: Parse  AVP 0, len 8, flag 0x8000 (M)
Tnl 21 L2TP: Parse SCCRP
Tnl 21 L2TP: Parse  AVP 2, len 8, flag 0x8000 (M)
Tnl 21 L2TP: Protocol Ver 1
Tnl 21 L2TP: Parse  AVP 3, len 10, flag 0x8000 (M)
Tnl 21 L2TP: Framing Cap 0x1
Tnl 21 L2TP: Parse  AVP 4, len 10, flag 0x8000 (M)
Tnl 21 L2TP: Bearer Cap 0x0
Tnl 21 L2TP: Parse  AVP 6, len 8, flag 0x0
Tnl 21 L2TP: Firmware Ver 0x500
Tnl 21 L2TP: Parse  AVP 7, len 14, flag 0x8000 (M)
Tnl 21 L2TP: Hostname BLIZZARD
Tnl 21 L2TP: Parse  AVP 8, len 15, flag 0x0
Tnl 21 L2TP: Vendor Name Microsoft
Tnl 21 L2TP: Parse  AVP 9, len 8, flag 0x8000 (M)
Tnl 21 L2TP: Assigned Tunnel ID 41
Tnl 21 L2TP: Parse  AVP 10, len 8, flag 0x8000 (M)
Tnl 21 L2TP: Rx Window Size 8
Tnl 21 L2TP: No missing AVPs in SCCRP
Tnl 21 L2TP: I SCCRP, flg TLS, ver 2, len 101, tnl 21, ns 0, nr 1 contiguous pak, size 101
C8 02 00 65 00 15 00 00 00 00 00 01 80 08 00 00
00 00 00 02 80 08 00 00 00 02 01 00 80 0A 00 00
00 03 00 00 00 01 80 0A 00 00 00 04 00 00 00 00
00 08 00 00 00 06 05 00 80 0E 00 00 00 07 42 4C
49 5A 5A 41 52 44 00 0F ...
Tnl 21 L2TP: O SCCCN  to BLIZZARD tnlid 41
Tnl 21 L2TP: O SCCCN, flg TLS, ver 2, len 20, tnl 41, ns 1, nr 1
C8 02 00 14 00 29 00 00 00 01 00 01 80 08 00 00
00 00 00 03
Vi1 Tnl/Sn 21/1 L2TP: O ICRQ to BLIZZARD 41/1
Vi1 Tnl/Sn 21/1 L2TP: O ICRQ, flg TLS, ver 2, len 48, tnl 41, lsid 1, rsid 1, ns 2, nr 1
C8 02 00 30 00 29 00 00 00 02 00 01 80 08 00 00
00 00 00 0A 80 08 00 00 00 0E 00 01 80 0A 00 00
00 0F 00 00 00 00 80 0A 00 00 00 12 00 00 00 02
Tnl 21 L2TP: O ZLB ctrl ack, flg TLS, ver 2, len 12, tnl 41, ns 3, nr 1
C8 02 00 0C 00 29 00 00 00 03 00 01
Tnl 21 L2TP: I ZLB ctrl ack, flg TLS, ver 2, len 12, tnl 21, ns 1, nr 2
Tnl 21 L2TP: I ZLB ctrl ack, flg TLS, ver 2, len 12, tnl 21, ns 1, nr 3
Vi1 Tnl/Sn 21/1 L2TP: Parse  AVP 0, len 8, flag 0x8000 (M)
Vi1 Tnl/Sn 21/1 L2TP: Parse ICRP
Vi1 Tnl/Sn 21/1 L2TP: Parse  AVP 14, len 8, flag 0x8000 (M)
Vi1 Tnl/Sn 21/1 L2TP: Assigned Call ID 1
Vi1 Tnl 21/1 L2TP: No missing AVPs in ICRP
Vi1 Tnl/Sn 21/1 L2TP: I ICRP, flg TLS, ver 2, len 28, tnl 21, lsid 1, rsid 1, ns 1, nr 3 contiguous
pak, size 28
C8 02 00 1C 00 15 00 01 00 01 00 03 80 08 00 00
00 00 00 0B 80 08 00 00 00 0E 00 01
Vi1 Tnl/Sn 21/1 L2TP: O ICCN to BLIZZARD 41/1
Vi1 Tnl/Sn 21/1 L2TP: O ICCN, flg TLS, ver 2, len 40, tnl 41, lsid 1, rsid 1, ns 3, nr 2
C8 02 00 28 00 29 00 01 00 03 00 02 80 08 00 00
00 00 00 0C 80 0A 00 00 00 18 00 98 96 80 80 0A
00 00 00 13 00 00 00 01
Tnl 21 L2TP: O ZLB ctrl ack, flg TLS, ver 2, len 12, tnl 41, ns 4, nr 2
C8 02 00 0C 00 29 00 00 00 04 00 02
Tnl 21 L2TP: I ZLB ctrl ack, flg TLS, ver 2, len 12, tnl 21, ns 2, nr 4
%UPDOWN: Interface Virtual-PPP1, changed state to up
%UPDOWN: Line protocol on Interface Virtual-PPP1, changed state to up
Ruijie# show vpdn
L2TP Tunnel and Session Information Total tunnels 1 sessions 1
LocID RemID Remote Name   State  Remote Address  Port  Sessions L2TP Class/
                                                      VPDN Group
21   41   BLIZZARD      est    192.168.12.213  1701  1
LocID     RemID     TunID     Username, Intf/     State  Last Chg
```

```
                           Vcid, Circuit
1        1        21        13,Vi1            est    00:00:13
%No active PPTP tunnels
Ruijie#
```

Maintaining L2TP Tunnels

Use the following command to clear a specified L2TP tunnel.

Command	Function
Ruijie# **clear vpdn tunnel** [{ **pptp** | **l2tp**} [*remote-host-name*]]	Clears a specified tunnel.

remote-host-name is the name of the peer host of a tunnel. In addition, all L2TP-related configuration commands in RGOS support instant configuration and use. Users can set or change L2TP parameters as required. The following is an example of clearing all tunnels.

```
Ruijie# show vpdn
L2TP Tunnel and Session Information Total tunnels 1 sessions 1
LocID RemID Remote Name   State  Remote Address  Port  Sessions L2TP Class/
                                                  VPDN Group
22   18889 C3640     est   192.168.12.242  1701  1       1

LocID   RemID   TunID    Username, Intf/    State  Last Chg
                           Vcid, Circuit
1     1280     22       ,Vi1           est   00:14:52
%No active PPTP tunnels
Ruijie# clear vpdn tunnel
Ruijie#
%UPDOWN: Line protocol on Interface Virtual-Access1, changed state to down
%CHANGED: Interface Virtual-Access1, changed state to administratively down
Ruijie# show vpdn
%No active L2TP tunnels
%No active PPTP tunnels
Ruijie#
```

FAQs

Common questions about L2TP tunnel establishment on the RGOS and their answers are presented as follows:

- Establishing an L2TP tunnel by interconnecting with a Windows PC: Note that L2TP is supported only in Microsoft Windows 2000 and later versions. Only the PPTP tunneling protocol is supported in earlier versions. Windows 2000 and Windows XP support L2TP by binding L2TP to IPSec/IKE. If you need to use a Windows 2000/XP PC to establish L2TP tunnels with other non-Microsoft network products, you must modify its registry to cancel this binding.
- Establishing an L2TP tunnel by interconnecting with a Windows PC: When you click **New connection** in **Network and Dial-up Connections** to create a virtual private connection (namely, **Connect to a private network through the Internet (V)**), the **Require data encryption (disconnect if none)** checkbox is selected by default and **Password requiring security measure** instead of PAP is used for authentication. However, the commoner method

is to use PAP or CHAP for authentication and not to encrypt data for transmission. Therefore, you need to modify the properties of this connection manually.

- Establishing an L2TP tunnel by interconnecting with a Windows PC: When you click **New connection** and **Accept incoming connections (A)** in **Network and Dial-up Connections** to create a connection, the MS-CHAPv2 method is used for user authentication by default and CHAP is forbidden for authentication. If you need to use CHAP for authentication, you must modify the configuration file of **Remote access policy** in the settings of the started **Routing and remote access** service, that is, add CHAP as an optional authentication method.
- Establishing an L2TP tunnel by interconnecting with a Windows PC: L2TP on Windows does not support the tunnel authentication function. The tunnel authentication function must be disabled in the L2TP settings of Ruijie router connected to the Windows PC. Tunnel authentication is disabled on Ruijie router by default.
- Establishing an L2TP tunnel by interconnecting with a Cisco device: Cisco IOS requires tunnel authentication by default, whereas tunnel authentication is diabled on Ruijie router by default.
- Establishing an L2TP tunnel by interconnecting with a Cisco device: If Ruijie router acts as the LNS and the Cisco device acts as the LAC to provide the L2TP tunnel service for remote dial-up users, **ip domain-lookup** instead of **ip cef** must be set on the Cisco device, just as on Cisco 2620 (LAC) in the preceding configuration example. Otherwise, the Cisco device will not forward PPP negotiation packets and other data to dial-up users.
- Establishing an L2TP tunnel by means of negotiation: The tunnel authentication settings at both ends of a tunnel must be consistent, that is, tunnel authentication is enabled or disabled at the same time at both ends. If it is enabled at both ends, the same tunnel authentication password must be set.
- Do not attempt to use Cisco routers (IOS versions 12.2 and 12.3) to establish an L2TP tunnel with a Windows 2000 PC, because it will be a waste of time.
- In consideration of being compatible with L2TP on different Windows versions and Cisco IOS of earlier versions, as well as the forwarding efficiency, RGOS does not support the AVP Hidden function and data message sequencing function. When an L2TP tunnel is etablished by interconnecting with Cisco IOS of later versions, ensure that the AVP Hidden function and data message sequencing function are disabled and default settings of the system are used.
- RGOS provides the LAC function, but does not support the LAC access server function.
- When RGOS is used to implement the LNS function, it is recommended that the address of the virtual-template interface be set in IP unnumbered mode, just as the settings on Cisco devices in consideration of fast route forwarding. Generally, IP unnumbered is bound to a loopback interface, as shown in the examples.
- In terms of instant configuration and use property of L2TP, effective changes (namely, non-repetitive operations) of control connection properties will cause the active disconnection of the related L2TP tunnel and all sessions on the tunnel. Effective changes of data transmission properties affects data transmission immediately.
- When the RGOS router is located behind a firewall, UDP port 1701 of the firewall must be enabled.
- When designing an L2TP tunnel, ensure that the route between the client and the server is available. When a router acts as the client, routes generated after an L2TP tunnel is established are different from those generated when a Windows PC acts as the client. If an L2TP tunnel is successfully established on a router (either a Cisco or a Ruijie product), two routes are generated, just as routes generated after other PPP link interfaces become UP. One is reachable to the server network segment, and the other is a direct route. However, when a Windows PC is used as the client, after an L2TP tunnel is established, a new route that traverses the L2TP tunnel and is reachable to the network 0.0.0.0/0 is added, in addition to the original route and the preceding two new routes.

Configuring VPDN 2.0

VPDN 2.0 Overview

VPDN 2.0, used to meet the need of VPDN with large capacity, has changed the configuration mode in VPDN server side.

The new virtual-vpdn interface configuration mode, which has replaced the original virtual-template one in VPDN server side, can meet the need of more tunnels' convergence.

Note

10.43b21 version support the function of VPDN 2.0, but only its L2TP tunnel mode. For the configuration in server side, only the configuration command related to interface has been changed. The original VPDN global parameter, and the configuration unrelated to interface have not. Users can continue to use them.

Configuring L2TP

Configuring Large Capacity L2TP

Configuration Task List

Configuring VPDN Address Pool (Optional)

This is an optional step for establishing an L2TP tunnel initiated by the remote client. In order to accept the L2TP connection initiated by the remote client, the LNS must allocate an IP address to the remote client if an IP address used inside the VPN is not set for the remote client. Generally, an idle IP address in a specified address pool is allocated to the client. Use the following commands to configure a VPDN address pool. VPDN module will manage the allocation by itself.

Command	Function
Ruijie(config)# **vpdn pool** *pool_name first-ip last-ip*	Creats or sets local address pool.
Ruijie(config)# **no vpdn pool** *pool_name*	Deletes the specified address pool.

poolname is the name of the local address pool to be created or set, *first-ip* is the first address in the address range set for the local address pool, and *last-ip* is the last address. When allocating address by using AAA, user can choose not to configure it. When we delete the address pool with "no" command, the corresponding binding command vpdn intf_pool on interface will also be deleted.

Configuring User Information

This is an optional step for establishing an L2TP tunnel initiated by the remote client. The purpose of configuring user information is to authenticate remote L2TP clients that attempt to access the local client.

Use the following commands to configure user information.

Command	Function
Ruijie(config)# **username** *user-name* **password {0|7}** *password*	Configures user information.
Ruijie(config)# **no username** *user-name*	Deletes the specified user.

user-name is the name of the dial-in user who is allowed to access, and *password* is the password of the user. The router locally maintains a database that records names of dial-in users permitted to access and their passwords.

Setting VPDN Global Parameters

This is a mandatory step for establishing an L2TP tunnel initiated by the remote client. VPDN global parameters are set in this step. The operations of setting VPDN global parameters include:

Enabling or Disabling the VPDN Function

If a user requires the router to accept the L2TP access from the remote client and establish an L2TP tunnel and session, the VPDN function must be enabled on the router.

Use the following commands to enable or disable the VPDN function.

Command	Function
Ruijie(config)# **vpdn enable**	Enables the VPDN function.
Ruijie(config)# **no vpdn enable**	Disables the VPDN function.

The VPDN enabling and disabling support instant configuration and use (that is, they are available immediately when they are configured). If the VPDN function is disabled, all the existing L2TP tunnels and sessions will be disconnected.

Setting the VPDN Source Address

After the VPDN source address is set, the destination address of the tunnel set for the remote client must be consistent with the VPDN source address before an L2TP tunnel is established successfully.

Use the following commands to set the VPDN (local) source address.

Command	Function
Ruijie(config)# **vpdn source-ip** *ip-address*	Sets the VPDN source address.
Ruijie(config)# **no vpdn source-ip** *ip-address*	Cancels the set VPDN source address.

The system does not check whether the destination address in the received tunnel establishment request is a specific value by default.

Setting the Maximum Number of VPDN Sessions

After the maximum number of VPDN sessions is set, access requests beyond the maximum value will be denied.

Use the following commands to set the maximum number of sessions supported by the VPDN server.

Command	Function
Ruijie(config)# **vpdn session-limit** *sessions*	Sets the maximum number of VPDN sessions.
Ruijie(config)# **no vpdn session-limit**	Restores the default maximum number of VPDN sessions.

The maximum number of sessions is the one configured by using this command by default.

Setting the Domain Resolution Option

Use the following commands to set the domain resolution option in VPDN domain authentication. Domains of different types can be identified based on configuration.

Command	Function
Ruijie(config)# **vpdn domain-delimiter @/%#-**	Sets the domain delimiter, prefix or suffix.
Ruijie(config)# **no vpdn domain-delimiter**	Cancels the VPDN domain authentication option.

The system does not resolve the domain field by default.

Enabling or Disabling Domain Authentication

Use the following commands to enable or disable the VPDN domain authentication, that is, determine whether to strip the domain field.

Command	Function
Ruijie(config)# **vpdn authorize domain split**	Enables domain authentication, that is, enables domain splitting.
Ruijie(config)# **no vpdn authorize domain**	Disables domain authentication.

Domain authentication is disabled by default.

Ignoring the Source Address Check of VPDN

Use the following commands to ingore errors on received L2TP control packets that do not comply with the RFC specifications so as to ensure the normal negotiation.

Command	Function
Ruijie(config-vpdn)# **vpdn ignore_source**	Ignores the source address check of packets sent from the peer end.
Ruijie(config-vpdn)# **no vpdn ignore_source**	Strictly checks the source address of packets sent from the peer end.

The system strictly checks the source address by default.

Setting VPDN Rate Limit

Use the following commands to limit the rate of establishing VPDN tunnels, namely, to limit the number of VPDN tunnels that can be established at one time.

Command	Function
Ruijie(config)# **vpdn limit_rate** *rate_num*	Enables rate limit. *rate_num* is the number of tunnels that can be established at a time. The value range is 5 to 100.
Ruijie(config)# **no vpdn limit_rate**	Disables rate limit.

Rate limit is disabled by default. When negotiating tunnel with large capacity, please enable the function so that to reduce shaking. When using virtual-template interface, please configure the number as 40. When using virtual-vpdn interface, please configure the number as 100.

Resolving Device Serial Number of VPDN

Use the following commands to set VPDN to resolve device serial number. According to the options, users can choose the division method. If the system has enabled LNS(L2TP) function, VPDN will resolve the device serial number transmitted from the peer side.

Command	**Function**
Ruijie(config)# **vpdn accept serial-number prefix**	Enables the function of resolving device serial number with prefix.
Ruijie(config)# **vpdn accept serial-number suffix**	Enables the function of resolving device serial number with suffix.
Ruijie(config)# **no vpdn accept serial-number**	Disables the function of resolving device serial number.

By default, the resolving device serial number function is disabled.

Transmitting Device Serial Number of VPDN

Use the following commands for to set VPDN to transmmit device serial number. If the system has enabled LNS(L2TP) function, VPDN will transmmit the device serial number to the peer side.

Command	**Function**
Ruijie(config)# **vpdn lac-send-devid**	Enables the function of transmmiting device serial number.
Ruijie(config)# **no vpdn lac-send-devid**	Disables the function of transmmiting device serial number.

By default, the transmmiting device serial number function is disabled.

Configuring a Virtual-Template Interface

This is a mandatory step for establishing an L2TP tunnel initiated by the remote client. This interface will become the logical interface that binds and carries L2TP sessions, and can only be used in LNS side. The operations of configuring a virtual-vpdn interface include:

> For information about setting the local IP address and setting the authentication mode see sections regarding the interface configuration guide. The virtual-vpdn interface can not configure ip unnumbered command, but can configure IP address and Mask directly.

Setting a Virtual-vpdn Interface

Use the following commands to set a virtual-vpdn interface. An interface can be binded with many sessions.

Command	Function
Ruijie(config)# **interface virtual-vpdn** *number*	Creates or configures a specified virtual-vpdn interface.
Ruijie(config)# **no interface virtual-vpdn** *number*	Deletes the specified virtual-vpdn interface.

number is the sequence number of the specified virtual-vpdn interface, and the range is <1-1024> . The created virtual-vpdn interface will be used as the logical interface that binds and carries L2TP sessions. When negotiating tunnels by using virtual-vpdn interface, the L2TP tunnel does not support IPV6 negotiation. All the PPP authentication of virtual-vpdn interface in vpdn-group's domain configuration need to be configured on the default virtual-vpdn interface of vpdn-group.

Setting VPDN Address Pool

Use the following commands to set the binding address pool on virtual-vpdn interface.

Command	Function
Ruijie(config)# **interface virtual-vpdn** *number*	Creates or configures a specified virtual-vpdn interface.
Ruijie(config-Virtual-vpdn *number*)# **vpdn intf_pool** *pool_name*	Binds the address pool on the interface.
Ruijie(config-Virtual-vpdn *number*)# **no vpdn intf_pool**	Deletes the binding address pool on the interface.

number is the sequence number of the specified virtual-vpdn interface, and the range is <1-1024>. *pool_name* is the name of the binding address pool. The VPDN binding address pool can only be used on the virtual-vpdn interface of L27P module.

Configuring VPDN-group

This is a mandatory step for establishing an L2TP tunnel initiated by the remote client. VPDN-group parameters are set in this step. The operations of configuring VPDN-group include:

Setting a VPDN-group Interface

Use the following commands to set a VPDN-group interface.

Command	Function
Ruijie(config)# **vpdn-group** *name*	Creates or configures the specified VPDN-group interface.
Ruijie(config)# **no vpdn-group** *name*	Deletes the specified VPDN-group interface.

name is the name of the specified VPDN-group interface. The created VPDN-group interface allows related clients to access and establish tunnels.

Setting the VPDN-group Source Address

The destination address in the tunnel establishment request sent from the remote client must match the VPDN-group source adress. In this way, the VPDN-group can be applied.

Use the following commands to set the VPDN-group source address.

Command	Function
Ruijie(config-vpdn)# **source-ip** *src-ip*	Sets the VPDN-group source address.
Ruijie(config-vpdn)# **no source-ip**	Cancels the set VPDN-group source address.

src-ip is the VPDN-group source address. When configuring with VPDN 2.0, the command is mandatory.

Setting the Tunneling Mode

Use the following commands to set the tunneling mode.

Command	Function
Ruijie (config-vpdn)# **accept-dialin**	Allows access from the dial-in remote client.
Ruijie (config-vpdn)# **no accept-dialin**	Denies the access from the dial-in remote client.

If a user needs the local router to provide the LNS function, access from the dial-in remote client must be allowed.

Setting the Tunneling Protocol

Use the following commands to set the tunneling protocol.

Command	Function
Ruijie(config-vpdn-acc-in)# **protocol** {**any** | **l2tp** | **pptp**}	Sets the tunneling protocol.
Ruijie(config-vpdn-acc-in)# **no protocol**	Cancels the set tunneling protocol.

The tunneling mode must be set before a tunneling protocol is set.

Setting a Logical Interface to Be Used

Use the following commands to set a logical interface to be used by VPDN-group,.

Command	Function
Ruijie(config-vpdn-acc-in)# **virtual-vpdn** *number*	Sets the using logical interface.
Ruijie(config-vpdn-acc-in)# **no virtual-vpdn** *number*	Cancels the set logical interface.

Before setting a logical interface used by VPDN-group, users should set the tunnel mode as L2TP and configure source-ip command in distributed system first. The logical interface can only be used in L2TP mode, and can not be used together with virtual virtual-template interface. All the PPP authentication of virtual-vpdn interface in vpdn-group's domain configuration need to be configured on the default virtual-vpdn interface of vpdn-group.

Setting the Peer Name

If the peer name is set, this VPDN-group is effective only to the remote client that matches the host name. If no peer name is set, this VPDN-group is the default VPDN-group and can provide the VPDN service for any remote client.

Use the following commands to set the peer name.

Command	Function
Ruijie(config-vpdn)# **terminate-from hostname** *name*	Sets the name of the peer host.
Ruijie(config-vpdn)# **no terminate-from**	Cancels the set peer name.

name is the name of the peer host.

Setting the Local Name

The local name is sent to the peer end as a record property.

Use the following commands to set the local name.

Command	Function
Ruijie(config-vpdn)# **local name** *name*	Sets the local name.
Ruijie(config-vpdn)# **no local name**	Cancels the set local name.

name is the local name. The router name is used as the local name and sent to the peer host of the tunnel by default.

Setting the L2TP Control Connection

Use the following commands to set the L2TP control connection.

Command	Function
Ruijie(config-vpdn)# **l2tp tunnel authentication**	Enables tunnel authentication.
Ruijie(config-vpdn)# **no l2tp tunnel authentication**	Disables tunnel authentication.
Ruijie(config-vpdn)# **l2tp tunnel hello** *interval*	Sets the interval of sending Hello messages.
Ruijie(config-vpdn)# **no l2tp tunnel hello**	Deletes the set interval of sending Hello messages.
Ruijie(config-vpdn)# **l2tp tunnel password** *pass-word*	Sets the tunnel password.
Ruijie(config-vpdn)# **no l2tp tunnel password**	Deletes the set tunnel password.
Ruijie(config-vpdn)# **l2tp tunnel receive-window** *size*	Sets the size of the receiving window of the tunnel control connection.
Ruijie(config-vpdn)# **no l2tp tunnel receive-window**	Restores the default size of the receiving window of the tunnel control connection.

Command	Function
Ruijie(config-vpdn)# **l2tp tunnel retransmit** {**retries** *number* | **timeout** {**min** | **max**} *seconds*}	Sets retransmission of tunnel control messages.
Ruijie(config-vpdn)# **no l2tp tunnel retransmit** {**retries** | **timeout** {**min**|**max**}}	Restores the default retransmission setting of tunnel control messages.
Ruijie(config-vpdn)# **l2tp tunnel timeout** {**no-session**| **setup**} *seconds*	Sets the maximum interval of establishing a no-session/control connection of the tunnel.
Ruijie(config-vpdn)# **no l2tp tunnel timeout** {**no-session** | **setup**}	Restores the default maximum interval of establishing a no-session/control connection of the tunnel.
Ruijie(config-vpdn)# **l2tp tunnel force_ipsec**	Enables forced encryption. It is used when external encryption is required. After this command is executed, packets can be sent to VPDN tunnels only after encryption.
Ruijie(config-vpdn)# **no l2tp tunnel force_ipsec**	Disables forced encryption.
Ruijie(config-vpdn)# **l2tp tunnel avp-hidden-compatible**	Supports the RFC2661 standard AVP Hidden parsing algorithm compatibly.
Ruijie(config-vpdn)# **no l2tp tunnel avp-hidden-compatible**	Restores the default Cisco standard AVP Hidden parsing algorithm.
Ruijie(config-vpdn)# **l2tp tunnel clear timeout** *seconds*	Sets the unix of vpdn-group's delete on time of session.
Ruijie(config-vpdn)# **no l2tp tunnel clear timeout**	Closes vpdn-group's delete on time function.

Tunnel authentication is not needed for establishing L2TP tunnels by default (but required on Cisco devices by default). The default interval of sending Hello control messages is 60 seconds. The default receiving window size of control messages is 4. The default number of retransmission times of control messages is 5. The default minimum and maximum intervals of retransmitting control messages are 1 second and 8 seconds respectively. The default maximum interval of no session in a tunnel is 600 seconds. The default maximum time supported by tunnels in establishing a control connection is 300 seconds. If L2TP tunnel authentication is required, the same tunnel password must be configured at both ends of the L2TP tunnel. Nevertheless, the system does not configure tunnel passwords for any L2TP tunnels by default and does not require tunnel authentication. The *interval* parameter is the interval of sending Hello messages, in seconds. The unit of the *seconds* parameter is also seconds. The forced IPSec encryption and authentication are disabled by default. The default AVP Hidden parsing algorithm uses the Cisco standard. After the RFC2661 standard AVP Hidden parsing algorithm is supported compatibly, the RFC2661 standard is used to parse and hide AVP. The interval of delete on time is 1 to 65535 minutes.

Setting L2TP Data Transmission Parameters

Use the following commands to set IP/UDP parameters for transmitting L2TP messages.

Command	Function
Ruijie(config-vpdn)# **l2tp ip udp checksum**	Sets the UDP checksum.

Ruijie(config-vpdn)# **no l2tp ip udp Checksum**	Cancels the UDP checksum setting.
Ruijie(config-vpdn)# **ip tos** *tos-value*	Sets the IP TOS field.
Ruijie(config-vpdn)# **no ip tos**	Cancels the IP TOS setting.
Ruijie(config-vpdn)# **ip precedence** *value*	Sets the IP Precedence field.
Ruijie(config-vpdn)# **no ip precedence**	Cancels the IP Precedence setting.

tos-value is the value of the TOS field of the IP header that carries L2TP messages, and *value* is the value of the Precedence field of this IP header. For L2TP messages to be carried in L2TP tunnels of RGOS, the checksum field of UDP that carries L2TP messages must be blank, the TOS of the IP header that carries L2TP messages and the Precedence field of this IP header must be 0 by default.

Note that the TOS and Precedence fields are supported only in L2TP. Though they can be configured in PPTP, the configuration does not take effect.

Setting the VRF Option

Use the following commands to set the VRF to which specified L2TP tunnel packets belong. The configuration maps to the **ip vrf forward** command of the VT interface, implementing VRF spanning.

Command	Function
Ruijie(config-vpdn)#**vpn vrf** *vrf-name*	Sets the VRF attribute for the tunnel.
Ruijie(config-vpdn)# **no vpn vrf**	Deletes the VRF attribute setting of the tunnel.

vrf-name is the name of the VRF.

Setting the Supported Domain Name

Use the following commands to set the domain name. The command for setting the supported domain name takes effect after the domain authentication is enabled. When the domain matching the content of this command is identified, put the configuration info of the domain into the L2TP session. If no matched domain info is found, deal with it by default.

Command	Function
Ruijie(config-vpdn)# **domain** *domain-name* **virtual-vpdn** *number*	Sets the authentication domain name and the related logical interface (when a virtual-vpdn logical interface is used).
Ruijie(config-vpdn)# **no domain** *domain-name*	Cancels the domain setting.

domain-name is the name of domain. *number* is the number of the logical interface, and the range is <1-1024>.

Re-Performing PPP Authentication

When the client triggers the LAC to start dialing, the LAC acts as the LNS to authenticate the client. This command is used to perform CHAP authentication on the client again after an L2TP tunnel is established. This commad is valid only on the LNS.

Use the following commands to forcibly perform complete PPP authentication again.

Command	Function
Ruijie(config-vpdn)# **force-local-chap**	Forces the LNS to perform CHAP authentication on the client again.
Ruijie(config-vpdn)# **no force-local-chap**	Cancels CHAP re-authentication.

Re-Performing PPP Negotiation

When the client triggers the LAC to start dialing, the LAC acts as the LNS to negotiate with the client. This command is used to perform LCP negotiation with the client again after an L2TP tunnel is established. This commad is valid only on the LNS.

Use the following commands to forcibly perform PPP negotiation again.

Command	Function
Ruijie(config-vpdn)# **force-local-lcp**	Forces the LNS to perform LCP negotiation with the client again.
Ruijie(config-vpdn)# **no force-local-lcp**	Cancels LCP re-negotiation.

Ignoring Errors on Control Packets

Use the following commands to ignore errors on received L2TP control packets that do not comply with the RFC specifications to ensure normal negotiation.

Command	Function
Ruijie(config-vpdn)# **lcp renegotiation always**	Ignores errors on packets from the peer end.
Ruijie(config-vpdn)# **no lcp renegotiation always**	Checks whether control packets comply with RFC specifications.

Setting the Domain Name’s Delete on Time of Session

Use the following commands to set the domain name’s delete on time of session. The command will be configured in vpdn-domain’s command mode.

Command	Function
Ruijie(config-vpdn)# **domain** *domain-name* **vrf** *vrf-name* Ruijie(config-vpdn-doamin)# **clear timeout** *seconds*	Sets the unix of domain name’s delete on time of session.
Ruijie(config-vpdn)# **no lcp renegotiation always**	Cancles the delete on time function.

domain-name is the name of domain, and *vrf-name* is the name of the VRF. Seconds is the interval of delete on time, and the range is 1 to 65535 minutes.

Setting No Sending of STOP Packet

The command is used to configure the behavior unmatched the RFC standard. It can only support one behavior, that is, do not send stop packet, when receiving SCCRQ repeatedly.

Command	Function
Ruijie(config-vpdn)# **l2tp tunnel none-rfc-compatible send-stop-pkt**	Does not send STOP packet, when receiving SCCRQ repeatedly
Ruijie(config-vpdn)# **no l2tp tunnel none-rfc-compatible send-stop-pkt**	Does send STOP packet, when receiving SCCRQ repeatedly

Setting SCCRP Packet's No Carrying of Tunnel Authentication ResponseAVP

After configuring tunnel authentication of LNS side in L2TP, if the accepted SCCRQ packet does not carry challenge AVP 11, the replied SCCRQ packet will not carry challenge response AVP13. The no command will close the function. The command can be compatible with SCCRQ's link detection function developed by other vendor, so that to avoid that after the configuring of tunnel authentication, the accepted SCCRQ packet with no carrying of authentication information will get back with all the information being 0.

Command	Function
Ruijie(config-vpdn)#**l2tp tunnel zxkeepalive-compatible**	If the accepted SCCRQ does not carry challenge AVP, the replied SCCRQ packet will not resend challenge resend AVP.
Ruijie(config-vpdn)#**no l2tp tunnel zxkeepalive-compatible**	If the accepted SCCRQ does not carry challenge AVP, the replied SCCRQ packet will resend challenge resend AVP with all the information being 0.

Setting Supported Flow-limit QOS by VPDN

Use the following commands to set VPDN's supporting of flow-limit QOS rule. It can be configured in vpdn-group to show that all the sessions in the group will obey the rule. It can also be configured in domain to show that all the sessions in the domain will obey the rule. The function can only use together with virtual-vpdn interface.

Command	Function
Ruijie(config-vpdn)# **flow-label** *flow-id*	Configures QOS rule in vpdn-group.
Ruijie(config-vpdn)#**no flow-label**	Deletes QOS rule.
Ruijie (config-vpdn-domain)# **flow-label** *flow-id*	Configures QOS rule in domain.
Ruijie (config-vpdn-domain)# **no flow-label**	Deletes QOS rule.

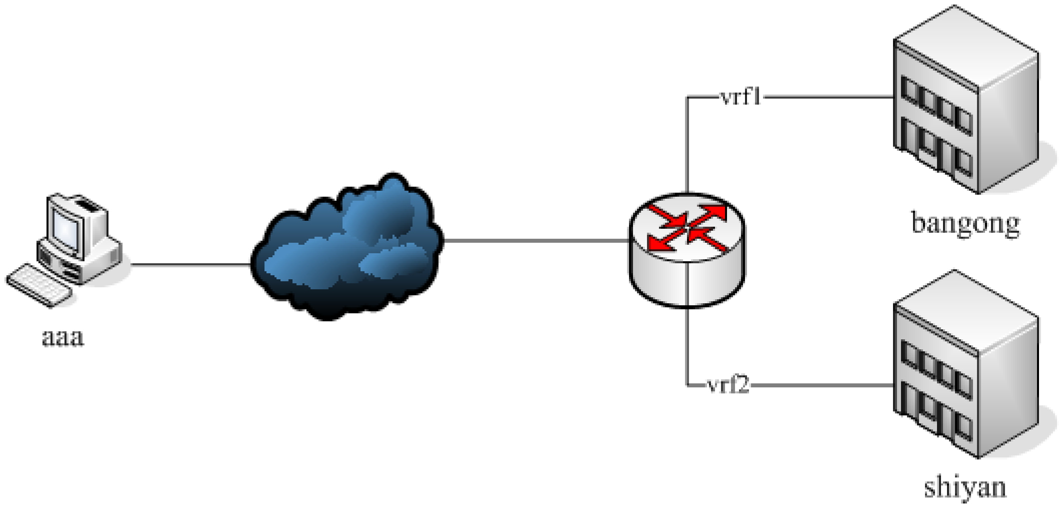
flow-id is the corresponding QOS rule's number, and its range is <1-1023>.

Configuration Examples

Configuration Example of L2TP Large Capacity and Domain Authentication

The following topology shows, according to domain information, a user dials in two networks by using one public network address.

Figure 28



```
ip vrf vrf1
ip vrf vrf2
!
vpdn enable
vpdn domain-delimiter @/%#-\ suffix
vpdn authorize domain split
vpdn pool vpdnusers 192.168.101.3 192.168.101.50
vpdn pool bangong 192.168.101.51 192.168.101.150
vpdn pool shiyan 192.168.101.151 192.168.101.253
!
!
username user@bangong password 7 025144391715
username user@shiyan password 7 127654431B
username user@ruijie password 7 164290823468
!
!
interface GigabitEthernet 1/1/1
ip address 192.168.19.1 255.255.255.0
duplex auto
speed auto
!
interface Virtual-vpdn 1
ppp authentication pap
ip address 192.168.101.2 255.255.255.0
```

```
vpdn intf_pool vpdnusers
!
interface Virtual-vpdn 2
ip vrf forward vrf1
ip address 192.168.101.2 255.255.255.0
ppp authentication pap
vpdn intf_pool bangong
!
interface Virtual-vpdn 3
ip vrf forward vrf2
ip address 192.168.101.2 255.255.255.0
ppp authentication pap
vpdn intf_pool shiyan
!
vpdn-group 1
! Default L2TP VPDN group
source-ip 192.168.19.1
accept-dialin
  protocol l2tp
  virtual-vpdn 1           /* bind logical interface */
domain bangong virtual-vpdn 2   /*specified domain*/
domain shiyan virtual-vpdn 3    /* specified domain */
!
!
line con 0
line aux 0
line vty 0 4
```

Sessions of dialing user who has no domain name will be binded with virtual-vpdn 1 interface. Sessions of dialing user whose domain name is bangong will be binded with virtual-vpdn 2 interface, and belongs to vrf1. Sessions of dialing user whose domain name is shiyan will be binded with virtual-vpdn 3 interface, and belongs to vrf2.

Configuration Example of Virtual-vpdn and OSPF

Different from virtual-access and virtual-ppp interface, virtual-vpdn interface can be binded with multichannel sessions. In this way, the configuration of its corresponding router protocol is also different.

LNS side：

```
username user1 password pass
!
vpdn enable
vpdn pool 10 76.1.11.1 76.1.11.100
vpdn pool 1 100.11.1.11 100.11.100.250
!
```

```
vpdn authorize domain split

vpdn domain-delimiter @ suffix

!

interface GigabitEthernet 2/1/5

 ip address 200.11.1.10 255.255.255.0

 duplex auto

 speed auto

!

interface Loopback 99

 ip address 99.9.9.9 255.255.255.0

!

interface Virtual-vpdn 1

 ppp authentication pap

 ip ospf network point-to-multipoint

 ip ospf source-check-ignore

 ip address 100.11.1.5 255.255.255.0

 vpdn intf_pool 1

!

!

vpdn-group 1

! Default L2TP VPDN group

 source-ip 200.11.1.10

 accept-dialin

  protocol l2tp

  virtual-vpdn 1

 domain a

!

!

router ospf 1

 network 99.9.9.0 0.0.0.255 area 0
 network 100.11.1.0 0.0.0.255 area 0
```

LAC side：

```
no service password-encryption
!
!
l2tp-class l2x1
!
pseudowire-class pw1
 encapsulation l2tpv2
 protocol l2tpv2 l2x1
!
interface GigabitEthernet 0/0
 ip address 200.11.1.250 255.255.255.0
 duplex auto
 speed auto
!
!
interface Loopback 7777
 ip address 77.7.7.7 255.255.255.0
!
interface Virtual-ppp 1
 ppp pap sent-username user1 password pass
 ip ospf source-check-ignore
 ip ospf hello-interval 30
 ip address negotiate
 pseudowire 200.11.1.10 1 encapsulation l2tpv2 pw-class pw1
!
!
router ospf 1
```

```
 network 77.7.7.0 0.0.0.255 area 0
 network 100.11.1.0 0.0.0.255 area 0
```

Configuration Example of Irtual-vpdn and RIP

LNS side:

```
username user1 password pass
!
vpdn enable
vpdn pool 10 76.1.11.1 76.1.11.100
vpdn pool 1 100.11.1.11 100.11.100.250
!
vpdn authorize domain split
vpdn domain-delimiter @ suffix
!
interface GigabitEthernet 2/1/5
 ip address 200.11.1.10 255.255.255.0
 duplex auto
 speed auto
!
interface Loopback 99
 ip address 99.9.9.9 255.255.255.0
!
interface Virtual-vpdn 1
 ppp authentication pap
 ip address 100.11.1.5 255.255.255.0
 vpdn intf_pool 1
!
vpdn-group 1
! Default L2TP VPDN group
 source-ip 200.11.1.10
 accept-dialin
```

```
  protocol l2tp
  virtual-vpdn 1
 domain a
!
router rip
 network 99.0.0.0
 network 100.0.0.0
 no validate-update-source
```

LAC side:

```
!
l2tp-class l2x1
!
pseudowire-class pw1
 encapsulation l2tpv2
 protocol l2tpv2 l2x1
!
interface GigabitEthernet 0/0
 ip address 200.11.1.250 255.255.255.0
 duplex auto
 speed auto
!
interface Loopback 7777
 ip address 77.7.7.7 255.255.255.0
!
interface Virtual-ppp 1
 ppp pap sent-username user1 password pass
 ip address negotiate
 pseudowire 200.11.1.10 1 encapsulation l2tpv2 pw-class pw1
!
```

```
router rip
 network 77.0.0.0
 network 100.0.0.0
 no validate-update-source
```

Monitoring and Maintaining VPDN 2.0

The methods of monitoring and maintaining VPDN 2.0 is just the same as the old one. Refer to the chapter of monitoring and maintaining L2TP.

Configuring the Tunnel Interface

Understanding the Tunnel interface

Overview

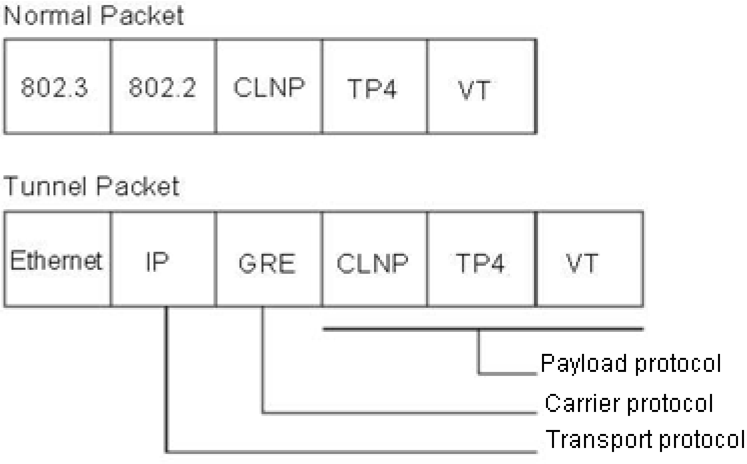
The tunnel interface is used to realize tunnel functions. Without specifically binding a certain transport protocol or payload protocol, the tunnel interface provides a standard point-to-point transmission link, and hence one tunnel interface must be configured for each separate link.

Tunnel function involves the following three key components:

34) Payload protocol: The protocol for encapsulating the payload (network data) transmitted through a tunnel. Currently, software of Ruijie products only support the use of the IP protocol as the payload protocol on the tunnel interface;
35) Carrier protocol: The protocol for secondary encapsulation and identification of the payload to be transmitted. Ruijie products support the following encapsulation modes on the tunnel interface: GRE and IPIP;
36) Transport protocol: The network protocol for transmitting the payload packets further encapsulated by the carrier protocol. Ruijie products use the most widely applied IP protocol as the transport protocol.

Figure shows the formation of a data packet encapsulated for transmission over an IP tunnel before and after the transmission.

Figure 29 The formation of data packet transmitted over the tunnel network before and after the transmission



IP tunnel transmission function is accomplished through GRE Ethernet encapsulation. In practice, if two private networks using the same protocol need to communicate with each other through the public network using a different protocol, they can use the tunnel function.

Tunnel transmission is applicable to the following circumstances:

37) Allowing the communication between non-IP local networks over a single-protocol network (IP network), as a tunnel supports different payload protocols; allowing the scope enlargement of a network running a hop-limited protocol;
38) Allowing the connection of discontinuous subnets over a single-protocol network (IP network);
39) Allowing the provision of VPN (virtual private network) over wide area network.

Since a tunnel will encapsulate the payload before transmission, such complexity in processing requires you to pay attention to the following issues under certain circumstances.

40) Since a tunnel is a point-to-point link which seems to have only one hop during routing, the actual routing overhead may involve multiple hops. Note that routing on a tunnel link may be different from the actual condition.
41) Since a tunnel will encapsulate the payload into a transport protocol, you need to give the corresponding consideration when configuring the firewall, especially the ACL. It shall also be noted that the transmission bandwidth (such as MTU) of a payload protocol is smaller than the theoretical value.

The followings will only introduce the attributes specific to the tunnel interface. The configuration of other attributes (including IP address and other relevant parameters, firewall and parameters of backup center) will be introduced in relevant sections.

Configuring the Tunnel Interface

Tunnel interface configuration tasks

Entering designated Tunnel interface configuration mode

To create a tunnel interface and enter interface configuration mode, run the following commands in global configuration mode:

Command	Function
Ruijie(config)# **interface tunnel** *tunnel-number*	Enters designated tunnel interface configuration mode.

Ruijie(config)# **no interface tunnel** *tunnel-number*	Deletes the existing tunnel interface.

Same as other logic interfaces, a tunnel interface is created once you enter the designated tunnel interface for the first time.

Configuring the source address of a Tunnel interface

A tunnel interface needs to identify the source address and destination address of the tunnel configured. In order to ensure the stability of a tunnel interface, the Loopback address is generally used as the source address and destination address of tunnel. Before normal operations of tunnel interface, check the connectivity between source address and destination address.

To configure the source address of a tunnel interface, run the following commands in tunnel interface configuration mode:

Command	Function
Ruijie(config-if)# **tunnel source** {*ip-address* | **interface-name** *interface-number*}	Configures the source address of the tunnel interface.
Ruijie(config-if)# **no tunnel source**	Removes the source address configuration of the tunnel interface.

The **tunnel source** command configures the actual source address for communication over a tunnel interface, namely the local endpoint of the tunnel.

Configuring the destination address of a Tunnel interface

To configure the destination address of a tunnel interface, run the following commands in tunnel interface configuration mode:

Command	Function
Ruijie(config-if)# **tunnel destination** { *ip-address* }	Configures the destination address of the tunnel interface.
Ruijie(config-if)# **no tunnel destination**	Removes destination address configuration of the tunnel interface.

The **tunnel destination** command configures the actual destination address for communication over a tunnel interface, namely the remote endpoint of the tunnel.

Caution

On the same router, tunnel interfaces that use the same encapsulation protocol cannot have the same source address or destination address.

Configuring Tunnel mode

The tunnel mode is referred to as the carrier protocol of a tunnel. The default tunnel mode is GRE. Of course, the user can also select a tunnel mode according to actual application.

Command	Function

Ruijie(config-if)# **tunnel mode** { **gre** {ip | ipv6}| **ipip** | ipv6ip}	Configures tunnel mode.
Ruijie(config-if)# **no tunnel mode**	Removes tunnel mode configuration and restores the default setting.

Caution The use of the **tunnel mode** command is related to device models.

Configuring Tunnel checksum

Under certain circumstances, tunnel checksum needs to be used to guarantee data integrity.

Command	**Function**
Ruijie(config-if)# **tunnel checksum**	Configures tunnel checksum.
Ruijie(config-if)# **no tunnel checksum**	Disables tunnel checksum.

By default, the checksum function of a tunnel interface is disabled.

Caution This command is only supported by the router.

Configuring the key of a tunnel interface

The key of a tunnel interface can ensure the security on both ends of a tunnel to a certain extent and prevent sniffing and attack from outside.

Command	**Function**
Ruijie(config-if)# **tunnel key** *key-value*	Configures the key of the tunnel interface.
Ruijie(config-if)# **no tunnel key**	Removes the key of the tunnel interface.

The key of a tunnel interface works only when the tunnel mode is GRE, as each GRE data packet will contain the tunnel key configured.

Caution (1) Both ends of a tunnel must use the same key configuration to allow normal communication;

(2) Although each GRE data packet will contain the key configured when the encapsulation mode is GRE, it is still unwise to guarantee security by relying on this key.

(3) This command is only supported by the router.

Configuring tunnel reception rules

If the payload protocol is inadequate to maintain the order of data packets, Ruijie products allow the configuration of tunnel reception rules to drop the disordered data packets. If the payload protocol is inadequate to maintain the order of data packets, this function can help realize the sequential transmission of data packets.

Command	Function
Ruijie(config-if)# **tunnel sequence-datagrams**	Configures sequential reception of packets on the tunnel.
Ruijie(config-if)# **no tunnel sequence-datagrams**	Removes sequential reception configuration of the tunnel.

This configuration is effective only when the tunnel mode is GRE.

Caution This command is only supported by the router.

Configuring TTL of a tunnel

Since a tunnel is a point-to-point link which seems to have only one hop during routing, the actual routing overhead may involve multiple hops. Ruijie products allow you to configure the TTL of a tunnel, namely to set the TTL in the transport protocol header of the packet transmitted over a tunnel. Being the intermediate node of tunnel, the router will reduce the TTL value in the transport protocol header and drop packets with TTL value being 0.

Command	Function
Ruijie(config-if)# **tunnel ttl** *hop-count*	Configures the TTL of the tunnel.
Ruijie(config-if)# **no tunnel ttl**	Removes TTL configuration of the tunnel and restores to the default value of 255.

By default, the TTL value of a tunnel transport protocol is 255.

Configuring TOS of a tunnel

In tunnel interface mode, configure the ToS byte of outer-layer transport protocol IPv4, or the 8 bits of traffic class of IPv6.

Command	Function
Ruijie(config-if)# **tunnel tos** *num*	Configures the TOS of the tunnel.
Ruijie(config-if)#**no tunnel tos**	Removes TOS configuration of the tunnel.

By default, if both the inner-layer carrier protocol and outer-layer encapsulation protocol of tunnel are IPv4, then the ToS byte of inner-layer IPv4 header will be copied to the outer-layer IPv4 header. If both the inner-layer carrier protocol and outer-layer encapsulation protocol of tunnel are IPv6, then the traffic class 8 bits of inner-layer IPv6 header will be copied to the outer-layer IPv6 header. In other cases, the outer-layer IPv4 ToS and IPv6 traffic class are 0.

Configuring PMTUD of a tunnel

Even if the payload IP message header is configured with DF (Don't Fragment) bit, the size of the payload protocol message may exceed the MTU of the destination outlet of tunnel after encapsulation, resulting in message fragmentation. On the way to the peer terminal of tunnel, the PMTU may become smaller and the intermediate forwarding device will fragment the encapsulated messages. Ruijie products provide the **tunnel path-mtu-discovery** command in interface configuration mode, allowing automatic discovery of PMTU in order to adjust the MTU size of tunnel interface and avoid message fragmentation.

Command	**Function**
Ruijie(config-if)# **tunnel path-mtu-discovery [age-timer** {*aging-mins* | **infinite} | min-mtu** *mtu-bytes]*	Enables the PMTUD function of the tunnel interface; Age-timer (optional): configures the aging timer of MTU on the tunnel interface; upon expiration of this timer, the MTU on the tunnel interface will reset to the initial MTU less the header length of carrier protocol messages; aging-mins: aging time ranging from 10 to 30 minutes, with default value being 10 minutes; infinite: disable MTU age-timer. Min-mtu (optional): configures MTU lower limit that can be adjusted by PMTUD; mtu-bytes: lower limit of MTU, ranging from 92 to 65535 bytes, with default value being 95 bytes.
Ruijie(config-if)# **no tunnel path-mtu-discovery [age-timer | min-mtu]**	Disables the PMTUD function of the tunnel interface

PMTUD can work only in GRE or IPIP tunnel mode, and is not enabled by default.

Caution

The PMTUD function requires the tunnel interfaces on both ends of the tunnel to be able to receive and process ICMP messages, especially when there is a firewall. This command is only supported by RGOS 10.4(2) or later versions.

After you run the **show interface tunnel** command, states of PMTUD are as follows:

Path MTU Discovery state:init

Path MTU Discovery state:keep

Path MTU Discovery state:learning

PMTUD learning has three state machines:

Initially, PMTUD is in init state.

When the timer expires and probe packets are being sent, PMTUD changes to the learning state and then learning packets are sent.

If an MTU change is not detected after five consecutive probe packets are sent, PMTUD changes to the keep state and begins sending keep packets.

Command	**Function**
Ruijie(config-if)# **tunnel path-mtu-discovery** *aging-mins mtu-bytes*	Enables the PMTUD function of the tunnel interface. Age-timer: configures the aging timer of MTU on the tunnel interface; upon expiration of this timer, the tunnel will send probe messages to discover Path MTU; aging-mins: aging time ranging from 1 to 65535 seconds. Min-mtu: configures MTU lower limit that can be adjusted by PMTUD; mtu-bytes: lower limit of MTU, ranging from 92 to 1500 bytes.

Ruijie(config-if)# **no tunnel path-mtu-discovery**	Disables the PMTUD function of the tunnel interface.

PMTUD can work only in GRE or IPIP tunnel mode, and is not enabled by default.

Caution The PMTUD function requires the tunnel interfaces on both ends of the tunnel to be able to receive and process ICMP messages, especially when there is a firewall. This command is only supported by RGOS 10.4(1) or later versions.

Configuring the keepalive function of a tunnel

When the physical interface sending tunnel messages is UP but the line failure prevents the tunnel messages from reaching the opposite terminal, the tunnel keepalive function can be used to detect the reachability of the tunnel interface.

Command	Function
Ruijie(config-if)# **keepalive** [*seconds* [*retries*]]	Configures the keepalive function of the tunnel.
Ruijie(config-if)# **no keepalive**	Disables the keepalive function of the tunnel

Caution This command is only supported by RGOS 10.4(2) or later versions.

This command cannot be used together with the **tunnel vrf** or **ip vrf forward** command.

This command applies only to GRE IP-capable 4 over 4 tunnels and IPIP tunnels.

Command	Function
Ruijie(config-if)# **tunnel keepalive** *period retries*	Configures the keepalive function of the tunnel.
Ruijie(config-if)# **no tunnel keepalive**	Disables the keepalive function of the tunnel

Caution This command is only supported by RGOS 10.4(1).

This command cannot be used together with the **tunnel vrf** or **ip vrf forward** command.

This command applies only to GRE IP-capable 4 over 4 tunnels and IPIP tunnels.

Configuring Tunnel nested encapsulation limit

Tunnel nested encapsulation refers to the circumstance that messages have undergone multi-level nested tunnel encapsulation on the local device before being sent out. The route change on the local device may result in infinite nested encapsulation. Excessive nesting will result in the continual fragmentation and recombination operations of the router and severely compromise routing performance. In order to avoid the occurrence of aforementioned phenomena, RGOS software can automatically avoid infinite nested encapsulation. Only 4-level nesting is allowed by default. Use the **tunnel nested-limit** command to modify the default value. This command is used on the tunnel interface of the innermost layer.

Command	Function
Ruijie(config-if)#**tunnel nested-limit** *num*	Configures the tunnel nested encapsulation limit. Default value: 4-level; value range: 0-10.
Ruijie(config-if)#**no tunnel nested-limit**	Restores the nested encapsulation limit to the default value.

Configuring Tunnel VRF

Identify which VRF would be used by the outer-layer transport protocol IPv4 for route selection and forwarding. Run the following commands:

Command	Function
Ruijie(config-if)# **tunnel vrf** *vrf-name*	Configures Tunnel VRF.
Ruijie(config-if)# **no tunnel vrf**	Removes Tunnel VRF configuration.

By default, the outer-layer IPv4 uses a global VRF table for route selection and forwarding. The source IP address and destination IP address of outer-layer encapsulation must be in the same VRF table. If in the designated VRF, there is no available route to the destination IP address, then this tunnel interface will be in down state.

Caution

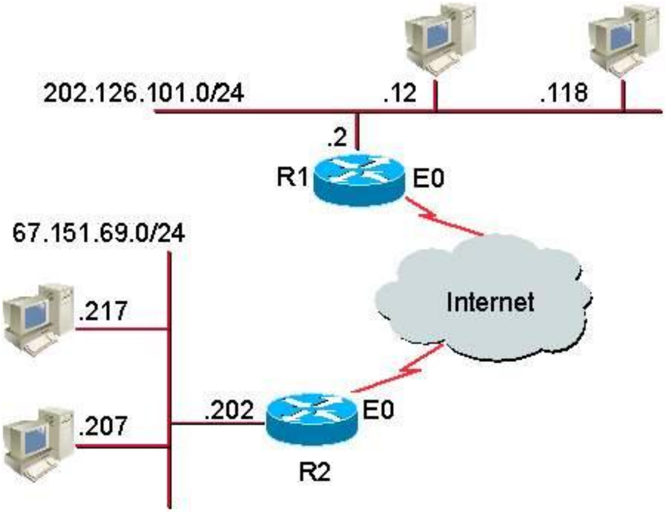
Currently, the tunnel VRF function can only support IPv4 over IPv4 GRE tunnel.

This command is only supported by RGOS 10.4(2) or later versions.

Example of Tunnel Interface Configuration

The network connections of this configuration example are shown in Figure .

Figure 30 Network connections of Tunnel interface configuration example



In the configuration example, a tunnel is created between R1 and R2. The subnet 202.126.101.0/24 behind R1 communicates with the subnet 67.151.69.0/24 behind R2 via the tunnel between R1 and R2. Such communication is carried out through the tunnel. The external network between R1 and R2 is transparent and invisible: a virtual private network (VPN). Tunnel configurations of R1 and R2 are shown below.

Configuration of R1:

```
interface Tunnel0
ip address 21.21.21.3 255.255.255.0
tunnel source 179.208.12.221
tunnel destination 179.208.12.55
!
interface FastEthernet0/0
ip address 179.208.12.221 255.255.255.0
!
interface FastEthernet0/1
ip address 202.106.101.2 255.255.255.0
!
```

Configuration of R2:

```
interface Tunnel0
ip address 21.21.21.5 255.255.255.0
tunnel source 179.208.12.55
tunnel destination 179.208.12.221
!
interface FastEthernet0/0
ip address 179.208.12.55 255.255.255.0
!
interface FastEthernet0/1
ip address 67.151.69.202 255.255.255.0
!
```

From the preceding configuration, you can learn that both R1 and R2 use Ethernet interface f0/0 to create a tunnel and use Ethernet interface f0/1 to connect to Intranet and serve as the gateway of Intranet.

Monitoring and Maintaining Tunnel interfaces

Ruijie products enable you to monitor and maintain tunnel interfaces by using the **show interfaces tunnel** and **debug [gre/ip | ipip]** commands.

Command	Function
Ruijie# **show interfaces tunnel** *tunnel-number*	Queries the status of a tunnel interface
Ruijie# **show tunnel gre**	Queries the general configurations of a GRE tunnel
Ruijie# **debug [gre/ip | ipip]**	Turns on tunnel debug switch.
Ruijie# **no debug [gre/ip | ipip]**	Turns off tunnel debug switch.

The examples show how to use **show interfaces tunnel** command and **debug** command.

42) Usage of the show interfaces tunnel command

```
Ruijie# show interfaces tunnel 1
Tunnel 1 is UP  , line protocol is UP
Hardware is  Tunnel
```

```
Interface address is: 1.1.1.1/24
MTU 1500 bytes, BW 9 Kbit
Encapsulation protocol is Tunnel, loopback not set
Keepalive interval is 0 sec , no set
Carrier delay is 0 sec
RXload is 1 ,Txload is 1
Tunnel source 192.168.200.200 (FastEthernet 0/0), destination 192.168.200.100
Tunnel protocol/transport GRE/IP, key 0xea
Order sequence numbers 0/0 (tx/rx)
Checksumming of packets enabled  Queueing strategy: WFQ
5 minutes input rate 0 bits/sec, 0 packets/sec
5 minutes output rate 0 bits/sec, 0 packets/sec
0 packets input, 0 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 abort
0 packets output, 0 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets
```

You can learn about the parameter settings and working status of the tunnel interface from the above information, such as the status of interface and link, IP address configuration, MTU configuration, bandwidth configuration and etc.

43) Usage of the **debug** command

```
Ruijie# debug gre/ip
Ruijie#
GRE: to decaps 192.168.200.100->192.168.200.200(len=132
ttl=255) 112
```

The above information indicates that a GRE/IP data packet is received from destination terminal (192.168.200.100) and de-encapsulated to obtain the IP payload packet.

Troubleshooting Faults on the Tunnel Interface

If both ends of a tunnel cannot communicate normally, carry out troubleshooting from the following aspects:

44) Make sure there is a reachable physical path between two ends of the tunnel, that is, the two ends are reachable for each other even if this tunnel is unavailable. Normal communication can be carried out between the source address (local terminal) of the tunnel and the destination address (peer terminal) of the tunnel.

45) Make sure the source address corresponds with the destination address, i.e., the source address must be identical with the destination address.

46) Make sure the tunnel uses the correct encapsulation mode (GRE by default), and both ends of the tunnel must use the same encapsulation mode.

47) After using GRE as the tunnel encapsulation protocol, make sure the checksum, Key and reception rule configurations are identical on both ends.

Configuring the AAA Function

Access control specifies the users who are allowed to access a server and lists the services that are accessible on the network. Authentication, authorization and accounting (AAA) is a key security mechanism for access control.

Overview

AAA presents a unified framework for configuring the authentication, authorization and accounting functions, which is supported by Ruijie products.

AAA provides the following services in a modular manner:

- Authentication: It verifies whether a user can get the right to access. User authentication is performed using RADIUS, TACACS+, or Local before a user accesses the network or a service on the network.
- Authorization: It determines the services which are accessible to a user by defining a series of attribute-value pairs (AVPs). These AVPs describe the operations the user is authorized to do. These AVPs can be stored on a network device or a remote RADIUS security server.
- Accounting: It records network resource usage of users. The network device starts sending resource usage of users to the Radius security server in the form of statistics when the accounting function is enabled. Every accounting record is stored in the security server as AVPs. These records can be read by special software to implement the accounting, statistics and tracing of network resource usage.

Note

Some products only provide the authentication function. For all problems with product specifications, contact the marketing or technical support personnel.

Although AAA is the primary access control method boasting superior security protection, Ruijie products also provide simpler control access methods, such as the local username authentication and line password authentication.

AAA has the following advantages:

- Excellent flexibility and controllability
- Expandability
- Standardized authentication
- Multiple backup systems

Basic AAA Principles

AAA types can be dynamically configured on a per-user (line) or per-server basis by creating method lists and applying them to specific services or interfaces.

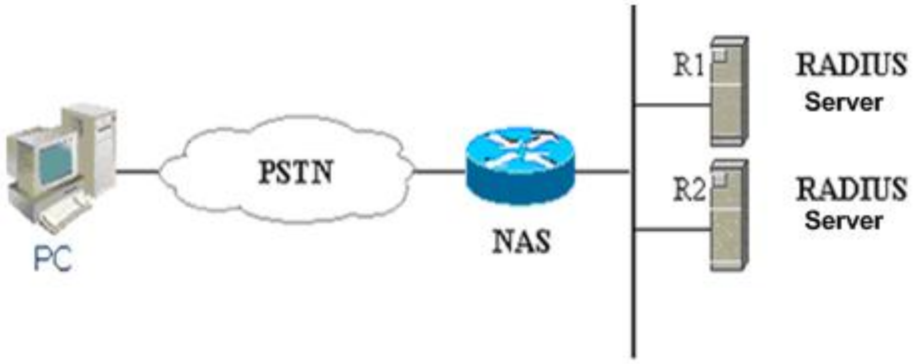
Method List

Since a variety of methods are available for user AAA, a method list should be used to define the sequence in which these methods are performed. The method list can define one or more security protocols for authentication, so that a backup system takes effect when the first method fails. In Ruijie products, a next method is selected if no response is received from the previous method till there is successful communication with a method or all methods in the list are attempted. If all methods listed are attempted but communication is not set up, AAA fails.

Caution

Only when there is no response from a method, Ruijie products will attempt the next method. During the authentication, if the user access is refused by a method, the authentication process ends and no other methods will be attempted.

Figure 31 Typical AAA network configuration



The preceding figure illustrates typical AAA network configuration, including two RADIUS security servers R1 and R2 and a network access server (NAS) that can function as a RADIUS server.

Supposed the system administrator has defined a method list, where user identity information is first obtained from R1, R2, and then the local username database on the NAS. If a remote PC user attempts to access the network via dialup, the NAS first queries the authentication information from R1. If the user is authenticated by R1, R1 sends a ACCEPT reply to the NAS, allowing the user to access the network. If R1 returns a REJECT reply, the user access is refused and connection from the user is rejected. If R1 does not reply, the NAS regards that timeout occurs and queries authentication information from R2. This process continues unless the user is authenticated, user access is rejected, or the session is terminated. If TIMEOUT is returned for all methods, the authentication fails and the user is disconnected.

Caution

The REJECT response is different from the TIMEOUT response. The server returns a REJECT message if a user fails to comply with the standard in the available authentication database. The server returns a TIMEOUT message if there is no response from the security server to the authentication. When an TIMEOUT message is detected, the next authentication method in the method list is selected to continue the authentication process.

Note

This document uses RADIUS for an example to describe the AAA function of security servers. For security access implementation based on TACACS+, see *Configuring TACACS+*.

AAA Configuration Steps

First you shall choose a security solution, evaluate the potential security risks in the specific network and select the proper measures to prevent unauthorized access. It is recommended that AAA be used to ensure network security.

AAA Configuration Description

AAA configuration may become simple when you understand the basic operation process of AAA. To configure AAA on network devices of Ruijie, perform the following steps:

48) Enable AAA by using the **aaa new-model** command in global configuration mode.
49) Configure parameters of the security protocol, RADIUS for example if you decide to use the security server.
50) Define the authentication method list by using the **aaa authentication** command.
51) Apply the method list to specific interface or line, if necessary.

Caution

When the specific method list is used, if no named method list is specified, the default authentication method list will apply.

As a result, if you do not want to use the default authentication method list, you shall define a specific method list.

For complete descriptions of the commands mentioned in this chapter, see related chapters in the *Security Configuration Command Reference*.

Enabling AAA

Before activating AAA security features, be sure to enable AAA.

To enable AAA, use the following command in global configuration mode:

Command	Function
Ruijie(config)# **aaa new-model**	Enables AAA.

Disabling AAA

To disable AAA, use the following command in global configuration mode:

Command	Function
Ruijie(config)# **no aaa new-model**	Disables AAA.

Follow-up Configuration

The following tables lists the possible configuration tasks that need to be completed after AAA enabling and chapters they are described in.

AAA access control security solution

Configuration task	Chapter
Configuring RADIUS Security Parameters	Configuring RADIUS
Configuring Local Login Authentication	Configuring Authentication
Defining AAA Authentication Method List	Configuring Authentication
Applying Method List to Specific Interface or Line	Configuring Authentication
Configuring RADIUS Authorization	Configuring Authorization
Enabling RADIUS Accounting	Configuring Accounting

If you are using AAA for authentication, see the *"Configuring Authentication" section*.

Configuring Authentication

Users need to be authenticated before they access network resources. In most cases, AAA is recommended for authentication.

Defining AAA Authentication Method List

To configure the AAA authentication, the first step is to define a named list of authentication methods, and then the applications use the defined method list for authentication. The method list defines the authentication types and sequence in which they are performed. The defined authentication methods, except the default method list, are specific to applications. Before a named method list is defined, all applications use the default method list.

A method list is simply a named list describing the authorization methods to be queried in sequence. Method lists enable you to designate one or more security protocols to be used for authorization, thus ensuring a backup system in case the initial method fails. Ruijie products use the first method listed to authorize users for specific network services; if that method fails to respond, Ruijie products select the next method listed in the method list. This process continues till there is successful communication with a listed authorization method, or all methods defined are exhausted.

⚠ Caution

Only when there is no response from a method, Ruijie products will attempt the next method. During the authentication, if the user access is refused by a method, the authentication process ends and no other methods will be attempted.

Configuration Examples

A typical AAA network has two RADIUS servers: R1 and R2. Suppose the network administrator has chosen a security solution, and the NAS authentication uses an authentication method to authenticate the Telnet connection. User authentication is initially attempted on R1, then on R2 if there is no response from, R1, and finally in the local database of the NAS if there is no response from R2 either. To design such an authentication process, you must configure an authentication method list accordingly by using the following commands:

Command	Function
configure terminal	Enters global configuration mode.
aaa authentication login default group radius local	Configures a default authentication method list named *default*. The protocols included in this method list are arranged behind the name according to the order in which they will be queried. The default method list applies to all applications by default.

To apply a method list to a specific Login connection, the system administrator must create a named method list and then apply it to the specific connection. The following example shows how to apply the authentication method list to line 2 only.

Command	Function
configure terminal	Enters global configuration mode.
aaa new-model	Enables AAA.
aaa authentication login test group radius local	Defines a method list named test in global configuration mode.
line vty *2*	Enters VTY line 2 configuration mode.
login authentication *test*	Applies the test method list to VTY line 2 in line configuration mode

If a remote PC user attempts to Telnet the NAS, the NAS first queries the authentication information from R1. If the user is authenticated by R1, R1 sends a ACCEPT reply to the NAS, allowing the user to access the network. If R1 returns a REJECT reply, the user access is refused and connection from the user is rejected. If R1 does not reply, the NAS regards that timeout occurs and queries authentication information from R2. This process continues unless the user is authenticated, user access is rejected, or the session is terminated. If both servers (R1 and R2) return TIMEOUT, the authentication will be performed by the local database of the NAS.

Caution

The REJECT response is different from the TIMEOUT response. The server returns a REJECT message if a user fails to comply with the standard in the available authentication database. The server returns a TIMEOUT message if there is no response from the security server to the authentication. When an TIMEOUT message is detected, the next authentication method in the method list is selected to continue the authentication process.

Authentication Type

Ruijie products support the following authentication types:

- Login authentication -- applies when a user tries to log in to the NAS through the command line interface (CLI).
- Enable authentication -- applies when an online user requests more rights on the CLI.
- PPP authentication -- applies to PPP dial-up users.
- DOT1X(IEEE802.1x) authentication -- applies to users who try to access through IEEE802.1x.

Configuring AAA Authentication

The following tasks are common for the configuration of AAA authentication.

- Enable AAA by using the **aaa new-model** command in global configuration mode.
- Configure the security protocol parameters if you decide to use the security server, such as RADIUS and TACACS+. See the "Configuring Radius" and "Configuring TACACS+" sections for details.
- Define the authentication method list by using the **aaa authentication** command.
- Applying the method list to a specific interface or line, if possible.

Caution TACACS+ is not supported by the DOT1X authentication on Ruijie products.

Configuring the AAA Login Authentication

This section describes how to configure the AAA Login authentication methods supported by Ruijie products:

Caution AAA security features can be made available only after AAA is enabled by using the **aaa new-model** command in global configuration mode. For the details, see the "AAA Overview" chapter.

In many cases, the user needs to Telnet the NAS for configuring the NAS remotely. To prevent unauthorized access to the NAS, user authentication is required.

The AAA security services make it easy for the network devices to perform line-based Login authentication. No matter which Login authentication method you use, you just need to use the **aaa authentication login** command to define one or more authentication method lists and apply them to the specific line that needs the Login authentication.

To configure the AAA Login authentication, run the following commands in global configuration mode:

Command	Function
configure terminal	Enters global configuration mode.
aaa new-model	Enables AAA.
aaa authentication login {default |*list-name} method1 [method2...]*	Defines an accounting method list. To define multiple method lists, repeat this command.
line vty *line-num*	Enters the line to which the AAA authentication applies.
login authentication {default|*list-name*}	Applies the method list to the line.

The keyword **list-name** is a character string used to name the created authentication method list, while **method** means the actual authentication algorithm. Only when the current method returns an ERROR message (no reply), the next authentication method will be attempted. If the current method returns a FAIL message, no authentication method will be used any more. To make sure that users can be authenticated even if no response is received from any method, use the **none** keyword.

In the following example, users can still be authenticated even if the RADIUS server returns TIMEOUT. Use the **aaa authentication login default group radius none** command.

Caution Since the keyword **none** enables every dial-up user to be authenticated even if the security server does not reply, it is used only as a backup authentication method. Normally, the **none** keyword is not recommended. You can use it as the last authentication method preceded by the local authentication method in the scenario where possible dial-up users are all trustful and their work are susceptible to any delay caused by system faults.

Keyword	Description
local	Uses the local username database for authentication.
none	User authentication is not performed.
group radius	Uses RADIUS to get authentication information.
group tacacs+	Uses TACACS+ to get authentication information.

The preceding table lists the AAA login authentication methods supported by Ruijie products.

Using the Local Database for Login Authentication

To use the local database for Login authentication, configure the local database first. Ruijie product supports authentication based on the local database. To enable the username authentication, run the following commands in global configuration mode:

Command	Function
configure terminal	Enters global configuration mode.
username *name* [**password** *password*]	Creates a local user and sets a password.
end	Returns to privileged mode.
show running-config	Verifies the configuration.

To define and apply the local login authentication method list, use the following commands:

Command	Function
configure terminal	Enters global configuration mode.
aaa new-model	Enables AAA.
aaa authentication login {default | *list-name*} **local**	Defines the local authentication method list.
end	Returns to privileged mode.
show aaa method-list	Verifies the configured method list.
configure terminal	Enters global configuration mode.
line vty *line-num*	Enters line configuration mode
login authentication {**default** | *list-name*}	Applies the method list.
end	Returns to privileged mode.
show running-config	Verifies the configuration.

Using Radius for Login Authentication

To use RADIUS for Login authentication, configure the RADIUS server. Ruijie products support the authentication based on the RADIUS server. To configure the RADIUS server, use the following commands in global configuration mode:

Command	Function
configure terminal	Enters global configuration mode.
aaa new-model	Enables AAA.

Command	Function
radius-server host *ip-address* [**auth-port** *port*] [**acct-port** *port*]	Configures the RADIUS server
end	Returns to privileged mode.
show radius server	Shows the RADIUS server.

After the RADIUS server is configured, make sure there is successful communication with the RADIUS server before configuring RADIUS for authentication. For details about the RADIUS server configuration, see the "Configuring RADIUS" section.

Then you can configure the RADIUS server based method list. Use the following commands:

Command	Function
configure terminal	Enters global configuration mode.
aaa new-model	Enables AAA.
aaa authentication login {**default** | *list-name*} **group radius**	Defines the local authentication method list.
end	Returns to privileged mode.
show aaa method-list	Verifies the configured method list.
configure terminal	Enters global configuration mode.
line vty *line-num*	Enters line configuration mode
login authentication {default | *list-name*}	Applies the method list.
end	Returns to privileged mode.
show running-config	Verifies the configuration.

Configuring the AAA Enable Authentication

This section describes how to configure the AAA Enable authentication methods supported by our product:

In many cases, the user needs to Telnet the NAS. After being authenticated, the user can access the CLI and is assigned 0–15 privilege levels initially, each having different commands. You can use the **show privilege** command to query the current level. For the details, see the "Using the CLI" section.

After logging in to the CLI, you can use the **enable** command to obtain higher privilege level if you fail to execute some commands. To prevent unauthorized access to the network, authentication needs to be performed when a user applies for a higher privilege level. This authentication type is called Enable authentication.

To configure the AAA Enable authentication, use the following commands in global configuration mode:

Command	Function
configure terminal	Enters global configuration mode.
aaa new-model	Enables AAA.
aaa authentication enable default *method1 [method2...]*	Defines an Enable authentication method list, for example RADIUS.

Only one Enable authentication method list can be defined globally, so there is no need to name the method list. The keyword **method** means the actual authentication algorithm. Only when the current method returns an ERROR message (no reply), the next authentication method will be attempted. If the current method returns a FAIL message, no authentication method will be used any more. To make sure that users can be authenticated even if no response is received from any method, use the **none** keyword.

Once configured, the Enable authentication method takes effect. When using **enable** command in privileged mode, the system prompts a message indicating authentication is required if you want to obtain a higher privilege level. There is no need to authenticate if the privilege level to be set is lower than or equal to the current one.

Caution

The current username will be recorded if the Login authentication(except for **none** method) is done when accessing the CLI. At this time, if the Enable authencation processes, a message indicating that the username must be entered will not be prompted and you can use the same username of Login authentication. Note that the password input must be consistent.

The username information will not be recorded if there is no Login authentication when you access the CLI, or the **none** method is used. At this time, if the Enable authentication is required, you shall enter the username again. This username will not be recorded, so you shall enter in each Enable authentication.

Some authentication methods can bind the security level. Then in the process of authentication, except for the returned response according to the security protocol, it is necessary to verify the bound security level. If the service protocol can bind the security level, the level shall be verified while authenticating. If the binded level is more than or equal to the level to be configured, the enable authentication and level switchover succeed. But if the bound level is less than the level to be configured, the Enable authentication fails, prompting an error message and keeping the current level. If the service protocol fails to be bound to the security level, you can configure the level without verification of the bound level.

Now only RADIUS and Local authentication can be bound to security levels. To this end, security levels need to be checked only for these two methods.

Using the Local Username Database for Enable Authentication

When configuring the local Enable authentication, you can configure the privilege level of local users. By default, the privilege level is 1. To configure the local Enable authentication, configure the local database and privilege levels. To enable the username authentication, use the following commands in global configuration mode:

Command	Function
configure terminal	Enters global configuration mode.
username *name* [**password** *password*]	Creates the local user and sets a password.
username *name* [**privilege** *level*]	Sets the user privilege level. (Optional)
end	Returns to privileged mode.
show running-config	Verifies the configuration.

To define the local Enable authentication method list, run the following commands:

Command	Function
configure terminal	Enters global configuration mode.
aaa new-model	Enables AAA.
aaa authentication enable default local	Defines the local authentication method list.
end	Returns to privileged mode.
show aaa method-list	Verifies the configured method list.

Command	Function
show running-config	Verifies the configuration.

Using RADIUS for Enable Authentication

The standard RADIUS server can pass the privilege level bound to the Service-Type attribute (the standard attribute number is 6), can specify the privilege with 1 or 15 level. The extened RADIUS server (for example, SAM) can configure the privilege level of the administrator (the private attribute number is 42), can specify 0-15 privilege level. For the details of the RADIUS server, see the "Specifying the RADIUS Private Attribute Type" section in "Configuring RADIUS".

To configure the RADIUS Enable authentication, configure the RADIUS server and then the RADIUS Enable authentication method list. Use the following commands in global configuration mode:

Command	Function
configure terminal	Enters global configuration mode.
aaa new-model	Enables AAA.
aaa authentication enable default group radius	Defines the RADIUS authentication method.
end	Returns to privileged mode.
show aaa method-list	Verifies the configured method list.
show running-config	Verifies the configuration.

Configuring the AAA Authentication for PPP Users

PPP is a link-layer protocol of carrying the network-layer datagram in the point-to-point link. In many circumstances, the user accesses the NAS by means of asynchronous or ISDN dail-up. Once the connection has been set up, the PPP negotiation will be enabled. To prevent the unauthorized access to the network, authentication is required for the dail-up user in the process of PPP negotiation.

This section describes how to configure the AAA Enable authentication methods supported by Ruijie products. To configure the AAA Enable authentication, use the following command in global configuration mode:

Command	Function
configure terminal	Enters global configuration mode.
aaa new-model	Enables AAA.
aaa authentication ppp {**default** | *list-name*} *method1 [method2...]*	Defines a PPP authentication method list. RADIUS, TACACS+ remote authentication and using the local database are the supported authentication methods.
interface *interface-type interface-number*	Enters the asynchronous or ISDN interface to which AAA authentication applies.
ppp authentication {chap | pap} {default | ***list-name***}	Applies the method list to the asynchronous or ISDN interface.

For details about PPP configuration, see the related chapter in *Configuring PPP and MP*.

Configuring the AAA Authentication for 802.1x Users

IEEE802.1x is a standard of Port-Based Network Access Control, providing the point-to-point secure access for the LAN, and a means of the authentication of the user connecting to the LAN device.

This section describes how to configure the 802.1x authentication methods supported by Ruijie products. To configure the AAA Enable authentication, use the following command in global configuration mode:

Command	Function
configure terminal	Enters global configuration mode.
aaa new-model	Enables AAA.
aaa authentication dot1x {**default** | *list-name*} *method1* *[method2...]*	Defines an IEEE802.1x authentication method list. RADIUS remote authentication and using the local database are the supported authentication methods.
dot1x authentication *list-name*	Applies the method list to 802.1x users.

For details about IEEE802.1x configuration, see the related chapter in *Configuring 802.1x*.

Example of Authentication Configuration

The following example illustrates how to apply both RADIUS authentication and local authentication to a network device.

```
Ruijie(config)# aaa new-model
Ruijie(config)# username Ruijie password starnet
Ruijie(config)# radius-server host 192.168.217.64
Ruijie(config)# aaa authentication login test group radius local
Ruijie(config)# line vty 0
Ruijie(config-line)# login authentication test
Ruijie(config-line)# end
Ruijie# show running-config
!
aaa new-model
!
!
aaa authentication login test group radius local
username Ruijie password 0 starnet
!
radius-server host 192.168.217.64
!
line con 0
line vty 0
login authentication test
line vty 1 4
!
!
```

In the preceding example, the access server uses the RADIUS server (IP address: 192.168.217.64) to perform Login authentication for users. If the RADIUS server does not reply, the local database will be used for authentication.

Example of Terminal Service Application Configuration

In the environment of the terminal service application, the terminal first connects to the asynchronous console, then offers the service accessing the network network server. However, if AAA is enabled, the Login authentication is necessary in all lines. To access the server, the terminal must pass the Login authentication and it influences the terminal service. You can separate two lines by configuration that makes the line using the terminal service directly connecting the server without the Login authentication, and ensures the device security by the Login authentication of the line connecting the device. That is to say, you can configure a login authentication list specific for the terminal service but the authentication method as **none**. Then apply the configured list to the line with terminal service enabled, while other lines connecting the local device is unchanged. In this way, the terminal can skip the local login authentication.

The following example illustrates the configuration steps:

```
Ruijie(config)# aaa new-model
Ruijie(config)# username Ruijie password starnet
Ruijie(config)# radius-server host 192.168.217.64
Ruijie(config)# radius-server key test
Ruijie(config)# aaa authentication login test group radius local
Ruijie(config)# aaa authentication login terms none
Ruijie(config)# line tty 1 4
Ruijie(config-line)# login authentication terms
Ruijie(config-line)# exit
Ruijie(config)# line tty 5 16
Ruijie(config-line)# login authentication test
Ruijie(config-line)# exit
Ruijie(config)# line vty 0 4
Ruijie(config-line)# login authentication test
Ruijie(config-line)# end
Ruijie(config)# show running-config
!
aaa new-model
!
!
aaa authentication login test group radius local
aaa authentication login terms none
username Ruijie password 0 starnet
!
radius-server host 192.168.217.64
radius-server key 7 093b100133
!
line con 0
line aux 0
line tty 1 4
login authentication terms
line tty 5 16
login authentication test
line vty 0 4
login authentication test
!
!
```

In the preceding example, the NAS uses the RADIUS server (IP address: 192.168.217.64) to perform login authentication for users. If the RADIUS server does not reply, the local database will be used for authentication. Login authentication is unnecessary for TTY 1-4 is the used line of the terminal service, while using other TTY and VTY lines needs the login authentication.

Configuring Authorization

The AAA authorization enables the administrator to control the use of services or rights. After the AAA authorization service is enabled, the network device configures the user sessions by using the user configuration file stored locally or in the server. After the authorization is completed, the user can only use the services allowed in the profile or has the assigned rights.

Authorization Types

Ruijie products support the following AAA authorization methods:

- Exec authorization: The user terminal loggs in to the CLI of the NAS and is granted the privilege level (0-15).
- Command authorization: After a user logs in to the CLI of the NAS, the user is specific commands are authorized.
- Network authorization: Grants the available service to the user session in the network.

Note

Only TACACS+ supports the command authorization method. For the detailed information, see the "Configuring TACACS+" section.

Preparations for Authorization

The following tasks must be completed before the AAA authorization is configured:

- Enable the AAA server. For details, see the AAA Overview chapter.
- (Optional) Configure the AAA authentication. The authorization is performed after the user is authenticated. But independent authorization can also be performed without authentication. For details of the AAA authentication, see the "Configuring Authentication section.
- (Optional) Configure security protocol parameters. If the security protocol is required for authorization, configure the security protocol parameters. The network authorization only supports RADIUS; the Exec authorization supports RADIUS and TACACS+. For details of the RADIUS, see the "Configuring RADIUS" section. For details of the TACACS+, see the "Configuring TACACS+ section.
- (Optional) If the local authorization is required, use the **username** command to define the user rights.

Configuring Authorization List

To enable AAA authorization, use the following commands in global configuration mode:

Command	Function
configure terminal	Enters global configuration mode.
aaa new-model	Enables AAA.
aaa authorization exec network{default | *list-name} method1 [method2*|*…*]	Defines the AAA Exec authorization method.

Command	Function
aaa authorization network network{**default** | *list-name} method1 [method2*|*…*]	Defines the AAA Network authorization method.

Configuring AAA Exec Authorization

The Exec authorization grants the privilege level of command execution for the user terminal logging in to the NAS. You can use the **show privilege** command to display the specific level after the user logs in to the NAS CLI successfully (by telnet, for example).

No matter which Exec authorization method you use, you just need to use the **aaa authorization exec** command to define one or more authorization method lists and apply them to the line that needs the Exec authorization.

To configure the AAA Exec authorization, use the following commands in global configuration mode:

Command	Function
configure terminal	Enters global configuration mode.
aaa new-model	Enables AAA.
aaa authorization exec network{**default** | *list-name} method1 [method2*|*…*]	Defines the AAA Exec authorization method. If you need to define multiple methods, execute this command repeatedly.
line vty *line-num*	Enters the line to which the AAA Exec authorization method is applied.
authorization exec {**default** | *list-name*}	Applies the method to the line.

The keyword **list-name** is a character string used to name the created authorization method list, while the keyword **method** means the actual authorization algorithm. Only when the current method returns an ERROR message (no reply), the next authorization method will be attempted. If the current method returns a FAIL message, no authorization method will be used any more. To make sure that users can be authorized successfully even if no response is received from any method, use the **none** keyword.

In the following example, the Exec authorication is still successful even if the RADIUS server returns TIMEOUT:

aaa authorization exec default group radius none

Keyword	Description
local	Uses the local username database for Exec authorization.
none	Exec authorization is not performed.
group radius	Uses RADIUS for Exec authorization.
group tacacs+	Uses TACACS+ for Exec authorization.

The preceding table lists the AAA Exec authorization methods supported by Ruijie products.

Caution

The exec authorization is always used together with the login authentication, and they can be applied to the same line at the same time. But note that it is possible to have different results of the authentication and the authorization towards the same user because they can use different methods and servers. If the Exec authorization fails, a user cannot access the CLI even though the login authentication of the user is successful.

Using the Local Username Database for Exec Authorization

To configure the local Exec authorization, configure the local database first. You can configure the privilege level of local users. By default, the privilege level is 1. Use the following commands in global configuration mode:

Command	Function
configure terminal	Enters global configuration mode.
username *name* [**password** *password*]	Creates a local user and sets a password.
username *name* [**privilege** *level*]	Sets the user privilege level. (Optional)
end	Returns to privileged mode.
show running-config	Verifies the configuration.

To define the local Exec authorization method list, use the following commands:

Command	Function
configure terminal	Enters global configuration mode.
aaa new-model	Enables AAA.
aaa authorization exec {**default** | *list-name*} **local**	Defines the local authorization method list.
end	Returns to privileged mode.
show aaa method-list	Verifies the configured method list.
configure terminal	Enters global configuration mode.
line vty *line-num*	Enters line configuration mode.
authorization exec {**default** | *list-name*}	Applies the method list.
end	Returns to privileged mode.
show running-config	Verifies the configuration.

Using RADIUS for Exec Authorization

To configure the RADIUS Exec authorization, onfigure the RADIUS server. For details about the RADIUS server configuration, see the "Configuring RADIUS" section.

After configuring the RADIUS server, the RADIUS authorization method list can be configured. Use the following commands in global configuration mode:

Command	Function
configure terminal	Enters global configuration mode.
aaa new-model	Enables AAA.
aaa authentication enable {default | *list-name*} **group radius**	Defines RADIUS authentication method.
end	Returns to privileged mode.
show aaa method-list	Verifies the configured method list.
configure terminal	Enters global configuration mode.
line vty *line-num*	Enters line configuration mode.
authorization exec {**default** | *list-name*}	Applies the method list.

Command	Function
end	Returns to privileged mode.
show running-config	Verifies the configuration.

Example of Configuring Exec Authorization

The following example illustrates how to configure Exec authorization. The local login authentication and the "Radius+local" Exec authorization are used when the user logs in through VTY lines 0-4. The NAS uses the RADIUS server with IP address set to 192.168.217.64 and shared keyword **test***.* The local username and password are *Ruijie,* and the privilege level is *6*.

```
Ruijie# configure terminal
Ruijie(config)# aaa new-model
Ruijie(config)# radius-server host 192.168.217.64
Ruijie(config)# radius-server key test
Ruijie(config)# username Ruijie password Ruijie
Ruijie(config)# username Ruijie privilege 6
Ruijie(config)# aaa authentication login mlist1 local
Ruijie(config)# aaa authentication exec mlist2 group radius local
Ruijie(config)# line vty 0 4
Ruijie(config-line)# login authentication mlist1
Ruijie(config-line)# authorization exec mlist2
Ruijie(config-line)# end
Ruijie(config)# show running-config
!
aaa new-model
!
aaa authorization lexec mlist2 group radius local
aaa authentication login mlist1 local
!
username Ruijie password Ruijie
username Ruijie privilege 6
!
Radius-server host 192.168.217.64
radius-server key 7 093b100133
!
line con 0
line vty 0 4
authorization exec mliat2
login authentication mlist1
!
end
```

Configuring AAA Network Authorization

Ruijie product support PPP and SLIP network authorization. The network authorization makes service configuration regarding traffic, bandwidth, and timeout available on the network connection. The network authorization only supports

RADIUS and TACACS+. The authorization information assigned by the server are encapsulated in the RADIUS attribute or TACACS+ attribute. Authorization information may vary with netowrk connections.

Caution Now AAA network configuration does not support 802.1X. For details about the 802.1X authorization, see the "Configuring 802.1X" section.

To configure the AAA network authorization, use the following commands in global configuration mode:

Command	Function
configure terminal	Enters global configuration mode.
aaa new-model	Enables AAA.
aaa authorization network{**default** | *list-name} method1 [method2*|*…*]	Defines an AAA network authorization method. If you need to define multiple methods, use this command repeatedly.

The keyword **list-name** is a character string used to name the created authorization method list, while **method** means the actual authorization algorithm. Only when the current method returns an ERROR message (no reply), the next authorization method will be attempted. If the current method returns a FAIL message, no authorization method will be used any more. To make sure that users can be authenticated even if no response is received from any method, use the **none** keyword.

Using RADIUS for Network Authorization

To configure RADIUS Network authorization, configure the RADIUS server. For details about the RADIUS server configuration, see the "Configuring RADIUS" section.

After configuring the RADIUS server, the RADIUS Network authorization method list can be configured. Use the following commands in global configuration mode:

Command	Function
configure terminal	Enters global configuration mode.
aaa new-model	Enables AAA.
aaa authentication network {**default** | *list-name*} **group radius**	Defines a RADIUS Network authorization method.

Example of Configuring Network Authorization

The following example illustrates how to configure Network authorization.

```
Ruijie# configure terminal
Ruijie(config)# aaa new-model
Ruijie(config)# radius-server host 192.168.217.64
Ruijie(config)# radius-server key test
Ruijie(config)# aaa authorization network test group radius local
Ruijie(config-line)# end
Ruijie(config)# show running-config
!
```

```
aaa new-model
!
aaa authorization network test group radius none
!
radius-server host 192.168.217.64
radius-server key 7 093b100133
!
```

Configuring Accounting

The AAA accounting function enables you to trace the services and network resources used by the user. After the accounting function is enabled, the NAS or router sends network access records of users to the RADIUS security server by means of AVP. You may use some analysis software to analyze these data to implement the billing, audit and tracing function for the user's activities.

Accounting Types

Ruijie products currently support the following accounting types:

- Exec accounting -- records the accounting information when users access or exit the CLI of the NAS.
- Command accounting – records the specific commands executed after the user logs in to the CLI of the NAS.
- Network accounting – records the related information on the user session in the network.

Note

The command accounting function supports only TACACS+. For details, see the "Configuring TACACS+" section.

Preparations for Accounting

The following tasks must be completed before the AAA accouting is configured:

- Enable the AAA server. For details, see the "AAA Overview" chapter.
- Define the security protocol parameters. It is required to configure the security protocol parameters for accouting. The network accounting only supports RADIUS; the Exec accounting supports RADIUS and TACACS+; the Command accounting supports TACACS+ only. For details of RADIUS, see the "Configuring RADIUS" section. For details of TACACS+, see the "Configuring TACACS+" section.
- (Optional) Configure the AAA authentication. Certain types of accounting (for example, Exec accounting) are performed after the user is authenticated. In some circumstances, the accouting can also be performed without authentication. For details about AAA authentication, see the "Configuring Authentication" section.

Configuring AAA Exec Accounting

The Exec accounting records the information when users access or exit the CLI of the NAS. When a user logs in and accesses the NAS CLI, it sends the accounting start information to the security server. When the user exits the CLI, it sends the accounting stop information to the server.

Caution

Exec accounting starts only after login authentication of the user is successful. If no login authentication or **none** authentication method has been configured, Exec accounting is not performed. If a user does not send no accounting start information to the security server when logging in, no accounting stop information will be sent when the user logs out.

To configure the AAA Exec accounting, use the following commands in global configuration mode:

Command	Function
configure terminal	Enters global configuration mode.
aaa new-model	Enables AAA.
aaa accounting exec {**default** | *list-name}* **start**-**stop** *method1 [method2*|*…*]	Defines the AAA Exec accounting method list. If you need to define multiple method lists, use this command repeatedly.
line vty *line-num*	Enters the line to which the AAA Exec accounting applies.
accounting exec {**default** | *list-name*}	Applies the method list to the line.

The keyword **list-name** is a character string used to name the created accounting method list, while the keyword **method** means the actual accounting algorithm. Only when the current method returns an ERROR message (no reply), the next accounting method will be attempted. If the current method returns a FAIL message, no accounting method will be used any more. To make sure that users can be authorized successfully even if no response is received from any method, use the **none** keyword.

Note

The keyword **start-stop** is used for the NAS to send the accounting information at the start and end of the network service to the security server.

Using the RADIUS for Exec Accounting

To configure RADIUS Exec acounting, configure the RADIUS server. For details about the RADIUS server configuration, see the "Configuring RADIUS" section.

After configuring the RADIUS server, the RADIUS accounting method list can be configured. Use the following commands in global configuration mode:

Command	Function
configure terminal	Enters global configuration mode.
aaa new-model	Enables AAA.
aaa accounting exec {**default** | *list-name*} **start**-**stop** **group radius**	Defines a RADIUS accounting method.
end	Returns to privileged mode.
show aaa method-list	Verifies the configured method list.
configure terminal	Enters global configuration mode.
line vty *line-num*	Enters the line configuration mode.
accounting exec {**default** | *list-name*}	Applies the method list.

Command	Function
end	Returns to privileged mode.
show running-config	Verifies the configuration.

Example of Configuring Exec Accounting

The following example illustrates how to configure Exec accounting. The local login authentication and the RADIUS Exec authorization are used when the user logs in through VTY lines 0-4. The IP address and shared key of the RADIUS server are 192.168.217.64 and *test* respectively. The local username and password both are *Ruijie*

```
Ruijie# configure terminal
Ruijie(config)# aaa new-model
Ruijie(config)# radius-server host 192.168.217.64
Ruijie(config)# radius-server key test
Ruijie(config)# username Ruijie password Ruijie
Ruijie(config)# aaa authentication login auth local
Ruijie(config)# aaa accouting exec acct start-stop group radius
Ruijie(config)# line vty 0 4
Ruijie(config-line)# login authentication auth
Ruijie(config-line)# accounting exec acct
Ruijie(config-line)# end
Ruijie(config)# show running-config
!
aaa new-model
!
aaa accounting exec acct start-stop group radius
aaa authentication login auth local
!
username Ruijie password Ruijie
!
radius-server host 192.168.217.64
radius-server key 7 093b100133
!
line con 0
line vty 0 4
accounting exec acct
login authentication auth
!
end
```

Configuring AAA Network Accounting

Network accounting records the accounting information about user sessions, including the numbers of packets and bytes, IP address and username. Now network accounting only supports RADIUS.

Note The format of RADIUS accounting information varies with the RADIUS security server. The contents of the accounting records may also vary with Ruijie product versions.

To configure the AAA network accounting, use the following commands in global configuration mode:

Command	Function
configure terminal	Enters global configuration mode.
aaa new-model	Enables AAA.
aaa accounting network{default | *list-name}* start-stop *method1 [method2*|…]	Defines the AAA network accounting method list. If you need to define multiple method lists, use this command repeatedly.

The keyword **list-name** is a character string used to name the created accounting method list, while the keyword **method** means the actual accounting algorithm. Only when the current method returns an ERROR message (no reply), the next accounting method will be attempted. If the current method returns a FAIL message, no accounting method will be used any more. To make sure that users can be authorized successfully even if no response is received from any method, use the **none** keyword.

Using RADIUS for Network Accounting

To configure RADIUS network acounting, configure the RADIUS server. For details about the RADIUS server configuration, see the "Configuring RADIUS" section.

After configuring the RADIUS server, the RADIUS accounting method list can be configured. Use the following commands in global configuration mode:

Command	Function
configure terminal	Enters global configuration mode.
aaa new-model	Enables AAA.
aaa accounting network {**default** | *list-name*} **start-stop group radius**	Defines a RADIUS accounting method.

Example of Configuring Network Accounting

The following example illustrates how to configure network authorization using RADIUS.

```
Ruijie# configure terminal
Ruijie(config)# aaa new-model
Ruijie(config)# radius-server host 192.168.217.64
Ruijie(config)# radius-server key test
Ruijie(config)# aaa accounting network acct start-stop group radius
Ruijie(config-line)# end
Ruijie(config)# show running-config
!
aaa new-model
!
```

```
aaa accounting network acct start-stop group radius
!
radius-server host 192.168.217.64
radius-server key 7 093b100133
!
```

Monitoring AAA users

To view the information of the current login users, use the following commands in privileged user mode:

Command	Function
show aaa user { *id* | all }	View the information of the current AAA user.

Configuring VRF-supported AAA Group

Virtual Private Networks (VPNs) provide a secure method for bandwidth share on the backbone networks of ISPs. One VPN is the collection of the shared routes. Users connect to the ISP network through one or multiple interfaces. The VPN routing table is also called VPN routing//forwarding(VRF) table. AAA can specify the VRF for each self-defined server group.

In global configuration mode, use the following commands to configure VRF for the AAA group:

Command	Function
configure terminal	Enters global configuration mode.
aaa new-model	Enables AAA.
aaa group server radius *gs_name*	Configures the RADIUS server group and enters server group configuration mode.
ip vrf forwarding *vrf_name*	Specifies the VRF for the group.
end	Returns to privilege mode.

Note VRF must be supported by Ruijie products.

Configuring Login Lockout for Failed Authentication

To prevent users from cracking passwords, use a command to specify the number of attempts. If the number of login attempts exceeds the limit, the user is locked and cannot log in again in a period.

In global configuration mode, use the following commands to configure login parameters:

Command	Function
configure terminal	Enters global configuration mode.
aaa new-model	Enables AAA.

Command	Function
aaa local authentication attempts *<1-2147483647>*	Configures the number of login attempt.
aaa local authentication lockout-time*<1-2147483647>*	Configures the time (in hours) in which a user is locked when the number of login attempts of the user exceeds the limit.
show aaa user lockout {all | user-name *<word>*}	Displays the list of locked users.
clear aaa local user lockout {all | user-name *<word>*}	Clears the lockout user list.
End	Returns to privilege mode.

Note

By default, the number of login attempts is 3 and the lockout time is 15 hours.

Configuring Domain Name-based AAA Service

This section is organized as follows::

- Overview
- Domain name-based AAA service configuration tasks
- Domain name-based AAA service configuration notes

Caution

The domain name-based AAA service is applied to the IEEE802.1x authentication service. For the detailed IEEE802.1x protocol configurations, see the "Configuring 802.1x" section.

Overview

In the multi-domain environment, one NAS can provide the AAA service for users in different domains. Due to the different user attributes (such as the username, password, service type, privilege, ect) in each domain, users need to be distinguished by setting domains and each domain is configured with a unique attribute set including the AAA service method list (RADIUS for example).

Note

Ruijie products support the following username formats:

1. userid@domain-name
2. domain-name\userid
3. userid.domain-name
4. userid

Users named in the format of "userid" belong to the default domain.

Basic principles for configuring the domain name-based AAA service are as follows:

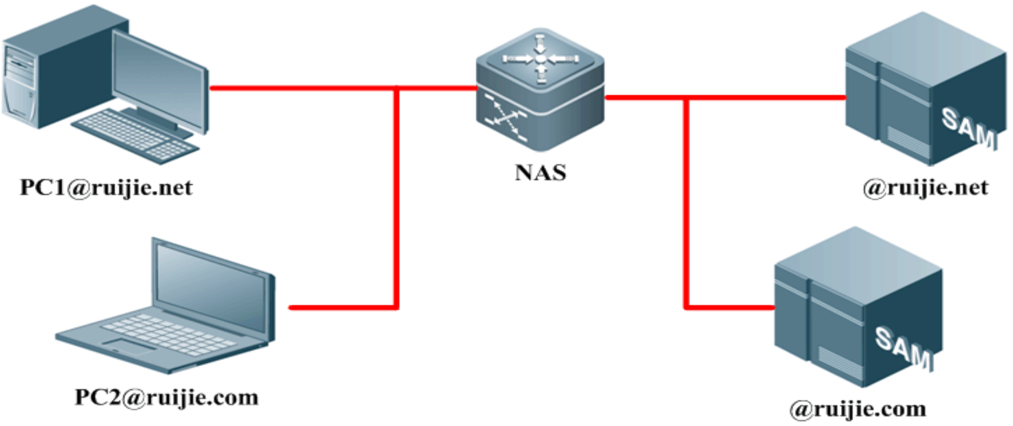
- Parsing the domain name of users
- Searching for the user domain according to the domain name

- Searching for the AAA service method list name according to the domain configurations
- Searching the corresponding method list according to the method list name in the system
- Providing the AAA service by using the method list

Note If one of the abovementioned steps fails, the AAA service cannot be used.

The following is the typical topology of a multi-domain environment:

Figure 32 Typical topology for a multi-domain network



Domain name-based AAA Service Configuration Tasks

Note The system supports up to 32 domains.

Enabling AAA

Command	Function
configure terminal	Enters global configuration mode.
aaa new-model	Enables AAA.

For detailed command descriptions, see the "Enabling AAA" section.

Defining the AAA Service Method List

Command	Function
configure terminal	Enters global configuration mode.
aaa authentication dot1x {**default** | *list-name*} *method1* [*method2*...]	Defines the IEEE802.1x authentication method list.
aaa accounting network {**default** | *list-name*} **start-stop** *method1* [*method2*...]	Defines the Network accounting method list.
aaa authorization network {**default** | *list-name*} *method1* [*method2*...]	Defines the Network authorization method list.

For detailed command descriptions, see the "Configuring authentication", "Configuring Accounting" and "Configuring authorization" sections..

Enabling the Domain Name-based AAA Service

Command	Function
configure terminal	Enters global configuration mode.
aaa domain enable	Enables the domain name-based AAA service.

Creating a Domain

You shall follow the following rules when searching for a domain by username:

52) A single character such as “.”, “\”, “@” can be used to distinguish between usernames and domain names.

53) The single “@” character is followed by the character string “domain-name”. With multiple “@” characters in the username, use the character string following the last “@” characeter as the domain-name. For example, if the username is a@b@c@d, use the a@b@c as the username and use the d as the domain-name.

54) The single “\” character follows the character string “domain-name”. With multiple “\” characters in the username, use the character string followed by the first “\” characeter as the domain-name. For example, if the username is a\b\c\d, use the b\c\d as the username and use the a as the domain-name.

55) The single “.” character is followed by the character string “domain-name”. With multiple “.” characters in the username, according to the pre-settings, use the character string following the last “.” characeter as the domain-name. For example, if the username is a.b.c.d, use the a.b.c as the username and use the d as the domain-name.

56) If all characters of “.”, “\” and “@” exist in the username, when matching the domain-name, use the rules in sequence of the “@”, “\” and “.” characters.

Command	Function
configure terminal	Enters global configuration mode.
aaa domain *domain-name*	Creates a domain and enters domain configuration mode.

Note

The AAA service supports domain names that have a maximum of 64 characters. Domain names are case insensitive.

Configuring the Domain Attribute Set

Use the following commands to select the AAA service method list in domain configuration mode:

Command	Function
authentication dot1x {**default |** *list-name*}	In domain configuration mode, select the authentication method list.
accounting network {**default |** *list-name*}	In domain configuration mode, select the accounting method list.
authorization network {**default |** *list-name*}	In domain configuration mode, select the authorization method list.

Use this command to configure the domain state:

Command	Function
state {block | active}	In domain configuration mode, set the domain state.

Use this command to check whether the username carries the domain name:

Command	Function
username-format {without-domain | with-domain}	In domain configuration mode, check whether the username carries the domain name information when the NAS is interacting with the server.

Use this command to set the maximum number of users supported in the domain:

Command	Function
access-**limit** *num*	In domain configuration mode, set the upper limit of users allowed in the domain. This function applies only to 802.1x users. By default, no upper limit is configured.

Note

1. Only AAA service method lists that have been configured can be selected in domain configuration mode. Otherwise, the system prompts that the AAA service method list you select does not exist.
2. With the domain name-based AAA service enabled, if there is no domain information carried by the username, use the default domain; if there is no configurations for the user domain in the system, the user is determined to be illegitimate and provides no AAA service.
3. In domain configuration mode, the default method list is selected if no other list is available.

Querying the Domain configuration

Use the following command to query the domain name-based AAA service information.

Command	Function
show aaa domain [*domain-name*]	Queries the current domain name-based AAA service information

Domain Name-based AAA Service Configuration Notes

When configuring the domain name-based AAA service, note the following points:

57) If the domain name-based AAA service is enabled, use the method list in the domain. If the service is not enabled, use the method list selected according to the access protocol (such as 802.1x, ect) for the AAA service. For example, if the service is not enabled, use the **dot1x authentication** *authen-list-name,* **dot1x accounting** *acct-list-name authen-list-name* and **dot1x accounting** *acct-list-name acct-list-name* command to provide the AAA service for the authentication and accouting method list name.
58) If the domain name-based AAA service is enabled, the default domain needs to be configured manually by default. The default domain is named "default" and is used to provide AAA services if the username does not contain domain

name. Without the default domain configured, the user whose name does not carry the domain information fails to use the AAA services.

59) If the domain information is carried by the auth-user but the domain is not configured on the device, it fails to provide the AAA service for the user.

60) The AAA service method list selected by the domain must be consistent with the one defined by the AAA service. Or it fails to provide the AAA service for the users in the domain.

61) The domain name carried by the user shall accurately match the one configured on the device. For example, the domain.com and the domain.com.cn have been configured on the device, and the request message carried by the user is aaa@domain.com, the device determines that the user belongs to the domain.com but not the domain.com.cn.

Domain Name-based AAA Service Configuration Example

The following is an example of configuring the domain name-based AAA service:

```
Ruijie(config)# aaa new-model
Ruijie(config)# radius-server host 192.168.197.154
Ruijie(config)# radius-server key test
Ruijie(config)# aaa authentication dot1x default group radius
Ruijie(config)# aaa domain domain.com
Ruijie(config-aaa-domain)# authentication dot1x default
Ruijie(config-aaa-domain)# username-format without-domain
```

After the configuration, with the user a1 in the radius server, use the 802.1x client to login the server for authentication by entering the username a1@domain.com and the correct password. The following shows the related domain name information:

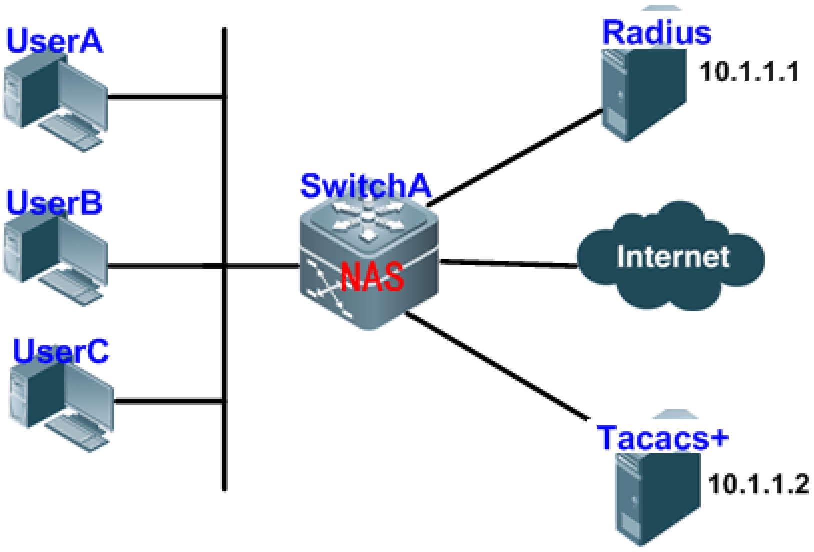
```
Ruijie#show aaa domain domain.com

=============Domain domain.com=============
State: Active
Username format: Without-domain
Access limit: No limit
802.1X Access statistic: 0

Selected method list:
 authentication dot1x default
```

Typical AAA Configuration Example

Typical AAA Aplication

Network Topology

Figure 33 Typical AAA Application Topology



Network Requirements

According to Figure 3, the following requirements must be met for better NAS security management:

62) The administrators shall have their own usernames and passwords, facilitating account management and preventing account leakage.

63) The user authentication methods are divided into local authentication and collection authentication. The method of combining the collection-authentication with the local-authentication shall be adopted, with the collection-authentication mainly-used and the local-authentication as backup. In the process of the collection-authentication, the Radius server authentication shall be passed first, if there is no reply, it will switch to the local authentication.

64) Different users can be configured to access the specified network device during the authentication.

65) Role-based management: Network management users are divided into the superusers and common users. Superusers have rights to query and configure the NAS, while common users only have limited query rights.

66) The user authentication information, the authorization information and the network information are recorded in the server for subsequent query and audit. (This example uses TACACS+ for accounting.)

Configuration Key-points

From the analysis of the part of "*Network Requirements*", deploying the AAA function can address the preceding requirements, which is to dynamically configure the ID authentication, authorization and accounting type for the user (line) or the server. Define the ID authentication, authorization and accounting type by creating the method list, and apply the method list to the specified service or interface. For details, see the "Configuration Steps" section.

Configuration Steps

#Enable AAA:

! Enable the AAA function on the device

```
Ruijie#configure terminal
Ruijie(config)#aaa new-model
```

# Configure the security server:

The security server provides the AAA services. Software of the server can record, calculate and analyze the various information in the form of logs.

! Configure the RADIUS server information (the shared key for the communication between the device and the RADIUS server is **ruijie**)

```
Ruijie(config)#radius-server host 10.1.1.1
Ruijie(config)#radius-server key ruijie
```

! Configure TACACS+ server information (the shared key for the communication between the device and the Tacacs+ server is **redgiant**)

```
Ruijie(config)#tacacs-server host 10.1.1.2
Ruijie(config)#tacacs-server key redgiant
```

# Configure the local user:

! Configure password encryption (the key information for the local password and the security server are saved and displayed in the simply-encrypted format).

```
Ruijie(config)#service password-encryption
```

! Configure the local user database (Configure the username and the password, and set the user privilege level).

```
Ruijie(config)#username bank privilege 10 password yinhang
Ruijie(config)#username super privilege 15 password star
Ruijie(config)#username normal privilege 2 password normal
Ruijie(config)#username test privilege 1 password test
```

! Configure the local enable password for the local Enable authentication.

```
Ruijie(config)#enable secret w
```

! Configure the line login password (It does not work when the AAA function is enabled. So the line login password configuration is to prevent the login failure with the AAA function disabled).

```
Ruijie(config)#line vty 0 15
Ruijie(config-line)#password w
```

! Configure the line user privilege level (with the Exec authorization disabled, or no Exec authorization method list is applied in the line and no default Exec authorization method list, the configure line user privilege level should be used).

```
Ruijie(config)#line vty 0 15
Ruijie(config-line)#privilege level 10
```

# Configure the authentication

1. Login authentication

The Login authentication is used to control the user access. There are two methods to define the authentication method list: 1) Radius; 2) Local.

! Configure login authentication method list and apply it to the corresponding line

```
Ruijie(config)# aaa authentication login hello group radius local
Ruijie(config)# line vty 0 15
Ruijie(config-line)# login authentication hello
```

To prevent the user from using the exhaust algorithm to crack the password during the Login authentication, AAA is used to limit the user Login attempts. When the the number of authentication attempts reaches the configured limit, the user is locked from login in a period (by default, three login authentication attempts are allowed and the lockout time is 15 hours.).

! Configure the number of allowed authentication attempts to 2 and the authentication lockout time to 10 hours

```
Ruijie(config)#aaa local authentication attempts 2
Ruijie(config)#aaa local authentication lockout-time 10
```

2. Enable authentication

The Enable authentication is used to switch the user privilege level. An authentication process is needed before the user switches the privilege level to the superuser using the **enable** command. There are two methods to define the authentication method list: 1) Radius; 2) Local. The Enable authentication can only set the default method list, which will be automatically applied after the configuration.

! Configure the enable authentication method list (RADIUS, TACACS+, and Local in descending order)

```
Ruijie(config)#aaa authentication enable default group radius local
```

# Configure the authorization

1. Exec authorization

The Exec authorization is used to control the user command privilege level. For example, level 15 is assigned to the superuser, level 14 is assigned to the configuration user, level 2 is assigned to the common user. The remote Exec authorization takes precedence over the local one.

! Configure the Exec authorization method list (TACACS+ has higher priority over Local) and apply it to the line

```
Ruijie(config)#aaa authorization exec shouquan group tacacs+ local
Ruijie(config)#line vty 0 15
Ruijie(config-line)#authorization exec shouquan
```

! Configure the exec authorization for the console (by default, the Exec authorization is not for the console)

```
Ruijie(config)#aaa authorization console
```

2. Command authorization

The Command authorization is used to offer the execution privilege of the key commands only to the administrators. The Command authorization authorizes the level of the command but not that of the current user. The RADIUS protocol is not supported.

! Configure the Command authorization method list (TACACS+ has higher priority over Local) and apply it to the line.

```
Ruijie(config)#aaa authorization commands 2 abc group tacacs+ local
Ruijie(config)#line vty 0 15
Ruijie(config-line)#authorization commands 2 abc
```

# Configure the accounting

1. Exec accounting

The Exec accouting is used to send the information about a user when the user accesses and exits the server for subsequent query, statistics, and audit.

! Configure the Exec accouting method list (TACACS+ accounting) and apply it to the line.

```
Ruijie(config)#aaa accounting exec default start-stop group tacacs+
```

2. Command accounting

The Command accouting is used to send the commands of a specific level executed by the user to the server for subsequent query, statistics and the audit.

! Configure the command accounting method list (TACACS+ only) and apply it to all lines.

```
Ruijie(config)#aaa accounting commands 2 default start-stop group tacacs+
```

Configuration verification

Step 1: Use the **show running-config** command to query the current configurations:

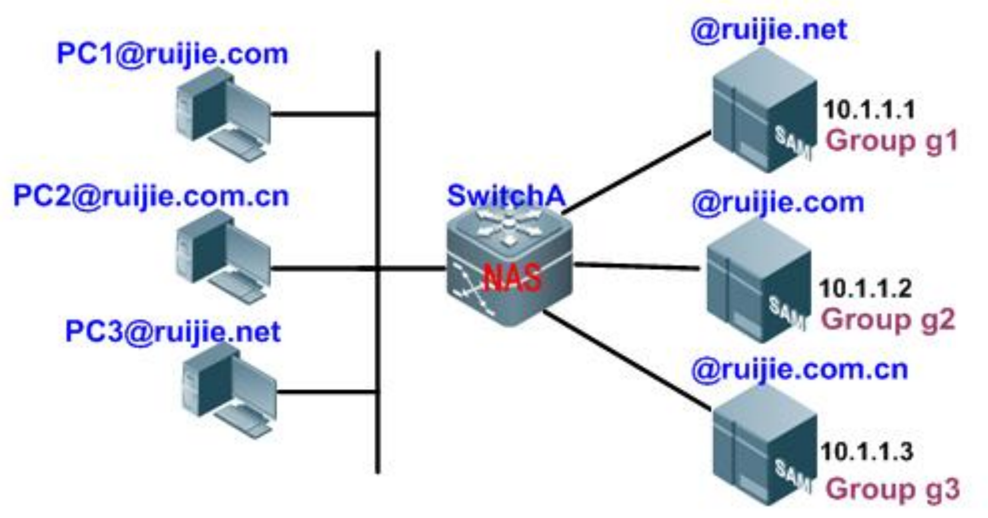
```
Ruijie(config)#show run

Building configuration...
Current configuration : 2337 bytes

!
!
aaa new-model
aaa local authentication attempts 2
aaa local authentication lockout-time 10
!
!
!
aaa authorization exec shouquan group tacacs+ local
aaa authorization commands 2 abc group tacacs+
aaa accounting exec default start-stop group tacacs+
aaa accounting commands 2 default start-stop group tacacs+
aaa authentication login hello group radius local
aaa authentication enable default group radius local
!
!
vlan 1
!
!
username bank password 7 09361c1c2f041c4d
username bank privilege 10
username super password 7 093c011335
username super privilege 15
username normal password 7 09211a002a041e
username normal privilege 2
username test password 7 093b100133
service password-encryption
!
!
!
!
tacacs-server key 7 072c062b121b260b06
tacacs-server host 10.1.1.2
radius-server host 10.1.1.1
radius-server key 7 072c16261f1b22
enable secret 5 $1$2MjW$xr1t0s1Euvt76xs2
!
!
!
!
!
line con 0
line vty 0 4
 authorization exec shouquan
 authorization commands 2 abc
 privilege level 10
 login authentication hello
 password 7 0938
line vty 5 15
 authorization exec shouquan
 authorization commands 2 abc
 privilege level 10
 login authentication hello
 password 7 005d
!
!
end
```

Step 2: In the actual application, use the **show aaa user {** *id* **| all }** command to query the current AAA user information.

AAA Multi-domain Authentication Application

Network Topology

Figure 34 AAA multi-domain authentication topology



Network Requirements

Configure the NAS device to enable the domain name-based AAA services:

- Use the 802.1x client for the login authentication with the username PC1@ruijie.com or PC2@ruijie.com.cn or PC3@ruijie,.net and the password.
- User network management: classify the users into superusers and common users, wherein the superusers are able to read and write while the common users are able to read only.
- The user authentication, authorization and network behavior are saved in the authentication server for subsequent query and audit.

Configuration Key Points

Configure the domain name-based AAA services to address the preceding network requirements.

The following example describes how to configure AAA multi-domain authentication on a 802.1x client.

Configuration Steps

#Enable AAA:

! Enable the AAA functions on the device

```
Ruijie#configure terminal
Ruijie(config)#aaa new-model
```

# Configure the security server:

The security server provides the AAA services. The user information is stored in the server and the software of the server can record, calculate and analyze the various information in the form of logs.

! Configure the RADIUS server information (the shared key for the communication between the device and the Radius server is **ruijie**)

```
Ruijie(config)#aaa group server radius g1
Ruijie(config-gs-radius)#server 10.1.1.1
Ruijie(config-gs-radius)#exit
Ruijie(config)#aaa group server radius g2
Ruijie(config-gs-radius)#server 10.1.1.2
Ruijie(config-gs-radius)#exit
Ruijie(config)#aaa group server radius g3
Ruijie(config-gs-radius)#server 10.1.1.3
Ruijie(config-gs-radius)#exit
Ruijie(config)#radius-server key ruijie
```

# Configure the local user:

! Configure the password encryption (the key information for the local password and the security server is saved and displayed in the simply-encrypted format).

```
Ruijie(config)#service password-encryption
```

! Configure the local user database (Configure the username and the password, and set the user privilege level).

```
Ruijie(config)#username bank privilege 10 password yinhang
Ruijie(config)#username super privilege 15 password star
Ruijie(config)#username normal privilege 2 password normal
Ruijie(config)#username test privilege 1 password test
```

! Configure the local Enable password for the local Enable authentication.

```
Ruijie(config)#enable secret w
```

# Define the AAA service method list

! Configure dot1x authentication.

```
Ruijie(config)#aaa authentication dot1x renzheng group radius local
```

! Configure network authorization.

```
Ruijie(config)#aaa authorization network shouquan group radius
```

! Configure network accounting.

```
Ruijie(config)#aaa accounting network jizhang start-stop group radius
```

# Enable the domain name-based AAA services

```
Ruijie(config)#aaa domain enable
```

# Create a domain and configure the domain attribute set

! Create a domain.

```
Ruijie(config)#aaa domain ruijie.com
```

! Associate the AAA service method list

```
Ruijie(config-aaa-domain)#authentication dot1x renzheng
Ruijie(config-aaa-domain)#authorization network shouquan
Ruijie(config-aaa-domain)#accounting network jizhang
```

! Configure the domain state.

```
Ruijie(config-aaa-domain)#state active
```

! Exclude the domain name from the username.

```
Ruijie(config-aaa-domain)#username-format without-domain
!
Ruijie(config)#aaa authentication dot1x renzheng group g2
Ruijie(config)#aaa authorization network shouquan group g2

Ruijie(config)#aaa accounting network jizhang start-stop group g2
```

The configurations of the ruijie.com.cn and the ruijie.net are similar.

Configuration Verification

Step 1: Use the **show running-config** command to query the current configurations (take the domain name **ruijie.com** for example):

```
Ruijie#show run

Building configuration...
Current configuration : 2013 bytes

!
aaa new-model
aaa domain enable
!
aaa domain ruijie.com
 authentication dot1x renzheng
 accounting network jizhang
 authorization network shouquan
 username-format without-domain
!
!
aaa group server radius g1
 server 10.1.1.1
!
aaa group server radius g2
 server 10.1.1.2
!
```

```
aaa group server radius g3
 server 10.1.1.3
!
!
aaa accounting network jizhang start-stop group g2
aaa authorization network shouquan group g2
aaa authentication dot1x renzheng group g2
!
!vlan 1
!
!
no service password-encryption
!
!
radius-server key ruijie
!
!
!

Step 2: Query the domain name-based AAA service domain information:
Ruijie#show aaa domain

=============Domain ruijie.com=============
State: Active
Username format: Without-domain
Access limit: No limit
802.1X Access statistic: 0

Selected method list:
 authentication dot1x renzheng
 authorization network shouquan


               accounting network jizhang
```

Configuring RADIUS

Overview of RADIUS

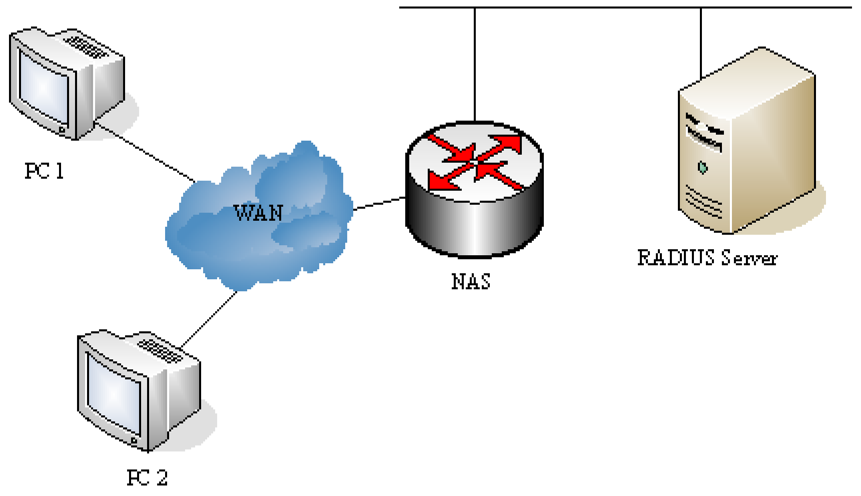
The Remote Authentication Dial-In User Service (Radius) is a distributed client/server system that works with the AAA to perform authentication for the users who are attempting to make connection and prevent unauthorized access. In the RGOS implementation, the RADIUS client runs on the router or the network access server (NAS) to send the authentication requests to the central RADIUS server. The central server stores all information of user authentication and network services.

Since RADIUS is a completely open protocol, it has become a component and been installed in such systems as Unix and Windows 2000, so it is the most popular security protocol for the time being.

The running process of RADIUS is as follows:

- Prompt the user to enter username and password.
- The username and the encrypted password are sent to the RADIUS server via the network.
- The RADIUS returns one of the following responses:
- ACCEPT: indicating that the user is authenticated.
- REJECT: indicating that the user authentication fails and the username and password must be entered again.
- CHALLENGE: indicating that the RADIUS server requests more authentication information from the user.
- The user authorization information is included in the ACCEPT response.

Figure 35 Typical RADIUS network



In addition to the authentication service, the RADIUS server also provides authorization and accounting services.

The RADIUS security protocol, also called the RADIUS method, is configured in the unit of a RADIUS server group. Every RADIUS method corresponds to a RADIUS server group which may consist of one or more RADIUS servers. For details about the RADIUS method, refer to *AAA-SCG.* If a RADIUS server group has multiple RADIUS servers, these RADIUS servers are used in polling mode till there is successful communication or all servers become unreachable.

RADIUS Configuration Tasks

To configure RADIUS on the network device, perform the following tasks first:

- Enable AAA. For the details, see *AAA-SCG*.
- Define a RADIUS authentication method list by using the **aaa authentication** command. For details about usage of the **aaa authentication** command, see the “Configuring Authentication” section.
- Apply the defined authentication method list to the specific line; otherwise the default authentication method list will be used for authentication. For more details, see the “Configuring Authentication” section.

Configuring RADIUS Protocol Parameters

Before configuring RADIUS on the network device, ensure that the RADIUS server is reachable. To configure RADIUS protocol parameters, run the following commands:

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **radius-server host** *ip-address* [**auth-port** *port*] [**acct-port** *port*]	Configures the IP address or hostname of the remote Radius security server and specifies the authentication port and accounting port.
Ruijie(config)# **radius-server key** *string*	Configures the shared key used for the communication between the device and the Radius server.
Ruijie(config)# **radius-server retransmit** *retries*	Specifies the times of sending a request before a RADIUS server is considered unreachable (3 by default).
Ruijie(config)# **radius-server timeout** *seconds*	Specifies the waiting time before the network device resends a request (5 seconds by default).

Caution

When configuring RADIUS, you must configure a RADIUS Key. Ensure that the network device and the RADIUS server use the same shared key.

Specifying Radius Authentication

This means defining the authentication method list for the Radius after the After specifying a RADIUS server and a RADIUS shared key, you must define a RADIUS authentication method list. RADIUS authentication is performed via AAA, so you need to run the **aaa authentication** command to define an authentication method list and specify the RADIUS authentication method. For more details, see *AAA-SCG*.

Specifying the Standard Radius Attribute Type

This section describes how to configure types of standard attributes. Now the RADIUS Calling-Station-ID attribute (the attribute value is 31) is supported.

Configuring Calling-Station-ID Format

The RADIUS Calling-Station-ID attribute is used to identify the NAS when the NAS is sending a request to the RADIUS server. The value of the RADIUS Calling-Station-ID is character strings, which can be in multiple formats. The MAC address for the NAS is usually used as the value of the Calling-Station-ID to solely identify the NAS. The table below describes the formats of the MAC address:

Format	Description
ietf	The standard format specified by IETF (in RFC3580). A hyphen (-) is used as the seperator, for example: 00-D0-F8-33-22-AC.
normal	Normal format of the MAC address (dotted hexadecimal format). A dot (.)is used as the seperator. For example: 00d0.f833.22ac.
unformatted	No format or seperator. By default, unformatted is used. For example: 00d0f83322ac.

To configure the format of the RADIUS Calling-Station-ID MAC-based attribute, run the following commands:

Command	Function
configure terminal	Enters global configuration mode.
radius-server attribute 31 mac format {ietf | normal | unformatted}	Configures the format of the RADIUS Calling-Station-ID MAC-based attribute. The default format is **unformatted**.

Specifying Private Radius Attribute Type

This section describes how to configure private attributes of RADIUS. By default, private RADIUS attributes are classified into Ruijie attributes and extended vendor types:

ID	Function	TYPE	Extended TYPE
1	max-down-rate	1	76
2	port-priority	2	77
3	user-ip	3	3
4	vlan-id	4	4
5	last-supplicant-version	5	5
6	net-ip	6	6
7	user-name	7	7
8	password	8	8
9	file-directory	9	9
10	file-count	10	10
11	file-name-0	11	11
12	file-name-1	12	12
13	file-name-2	13	13
14	file-name-3	14	14
15	file-name-4	15	15
16	max-up-rate	16	16
17	current-supplicant-version	17	17
18	flux-max-high32	18	18

ID	Function	TYPE	Extended TYPE
19	flux-max-low32	19	19
20	proxy-avoid	20	20
21	dailup-avoid	21	21
22	ip-privilege	22	22
23	login-privilege	42	42
26	ipv6-multicast-address	79	79
27	ipv4-multicast-address	87	87
62	sdg-type	62	62
85	sdg-zone-name	85	85
103	sdg-group-name	103	103

Note

Some private attributes are supported only by specific products. You can run the **show radius vendor-specific** command to view private attribute lists supported by products.

Two attributes cannot be configured with the same type number.

The following is an example about private attributes of network devices:

```
Ruijie# show radius vendor-specific
id   vendor-specific    type-value
----- -------------------- ----------
1   max-down-rate     76
2   port-priority     77
3   user-ip       3
4   vlan-id       4
......
Ruijie# configure
Ruijie(config)# radius attribute 4 vendor-type 67
Ruijie(config)# show radius vendor-specific
id   vendor-specific    type-value
----- -------------------- ----------
1   max-down-rate     76
2   port-priority     77
3   user-ip       3
4   vlan-id       67
......

Ruijie(config)#
```

Configuring RADIUS Server Reachability Detection

The device maintains the reachability state of each RADIUS server configured: reachable or unreachable. The device does not send authentication, authorization and accounting requests of users to an unreachable RADIUS server, unless all RADIUS servers in the RADIUS server group are unreachable.

The device can carry out proactive detection of the specified RADIUS server, and this feature is disabled by default. If you enable proactive detection of the specified RADIUS server, the device will periodically send detection requests (authentication requests or accounting requests) to the RADIUS server at an interval of:

- 60 minutes (the default value) for reachable RADIUS servers
- 1 minute (a constant value) for unreachable RADIUS servers

Note

To enable proactive detection of the specified RADIUS server, the following conditions must be met:

1. The test user name for this RADIUS server has been configured on the device.
2. At least one tested port of this RADIUS server (authentication port or accounting port) has been configured on the device.

For a reachable RADIUS server, the device will consider this RADIUS server unreachable if the following two conditions are met:

1. The time configured by using the **radius-server dead-criteria time** *seconds* command has elapsed since the receipt of the last correct response from the RADIUS server.
2. After the receipt of the last correct response from the RADIUS server, the number of requests (including retransmitted requests) without a response reaches the value configured by using the **radius-server dead-criteria tries** *number* command.

For an unreachable RADIUS server, the device will consider this RADIUS server reachable if any of the following conditions is met:

67) A correct response is received from this RADIUS server.
68) The duration that this RADIUS server remains unreachable exceeds the time set by using the **radius-server deadtime** command, and proactive detection of this RADIUS server is not enabled.
69) The authentication port or accounting port of this RADIUS server is updated on the device.

RADIUS server reachability detection allows the user to judge whether a RADIUS server is unreachable and to configure proactive detection.

To configure RADIUS server reachability detection, run the following commands in global configuration mode:

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **radius-server dead-criteria time** *seconds* **tries** *number*	Configures global criteria for judging whether a RADIUS server is reachable. The default value of *seconds* is **60**, and the default value of *number* is **10**.
Ruijie(config)# **radius-server deadtime** *minutes*	Configures the duration for the device to stop sending request packets to the RADIUS server in unreachable state (default value: 0 minutes).

Command	Function
Ruijie(config)# **radius-server host** *ip-address* [**auth-port** *port*] [**acct-port** *port*] [**test username** *name* [**idle-time** *time*] [**ignore-auth-port**] [**ignore-acct-port**]]	Configures the IP address of a remote RADIUS server, specifies the authentication port and accounting port, and specify relevant parameters of proactive detection (testing user name, interval for proactive detection of reachable RADIUS servers, and whether the authentication port or the accounting port shall be neglected).

Caution

In the configuration, a special testing user name shall be used. This user name cannot be used by other authorized users, avoiding adverse impact on authentication, authorization or accounting of these users.

Monitoring RADIUS

To monitor RADIUS, run the following command in privileged user mode:

Command	Function
debug radius { **event** | **detail** }	Turns on the Radius debug switch to view the Radius debug information.

Radius Configuration Example

In a typical Radius network configuration diagram, the RADIUS server performs authentication for the users who are attempting to access, enables the accounting function for these users and records the network service usage of them.

Note

The RADIUS server can be a component that comes with the Windows 2000/2003 server (IAS) or the Unix system, or special certified server software of some manufacturers.

The following example shows how to configure the Radius on the network device:

```
Ruijie# configure terminal
Ruijie(config)# aaa new-model
Ruijie(config)# radius-server host 192.168.12.219 auth-port 1645 acct-port 1646
Ruijie(config)# radius-server key aaa
Ruijie(config)# aaa authentication login test group radius
Ruijie(config)# end
Ruijie# show radius server
Server IP:   192.168.12.219
Accounting Port: 1646
Authen Port:  1645
Test Username:  <Not Configured>
Test Idle Time:  60 Minutes
Test Ports:   Authen and Accounting
Server State:  Active
```

```
    Current duration 765s, previous duration 0s
  Dead: total time 0s, count 0
  Statistics:
   Authen: request 15, timeouts 1
   Author: request 0, timeouts 0
   Account: request 0, timeouts 0

Ruijie# configure terminal
Ruijie(config)# line vty 0
Ruijie(config-line)# login authentication test
Ruijie(config-line)# end
Ruijie# show running-config
!
aaa new-model
!
!
aaa authentication login test group radius
!
!
!
radius-server host 192.168.12.219 auth-port 1645 acct-port 1646
radius-server key aaa
!
line con 0
line vty 0
login authentication test
line vty 1 4
!
```

RADIUS IPv6 Configuration Example

In the typical RADIUS network configuration diagram, the RADIUS server performs authentication and accounting of users, and records the network service service usage of them.

Note

The RADIUS server is deployed on a Windows 2008 Server or special IPv6 capable server software certified by manufacturers.

The following example shows how to configure RADIUS on the network device:

```
Ruijie# configure terminal
Ruijie(config)# aaa new-model
Ruijie(config)# radius-server host 3000::100 auth-port 1645 acct-port 1646
Ruijie(config)# radius-server key aaa
Ruijie(config)# aaa authentication login test group radius
Ruijie(config)# end
Ruijie# show radius server
Server IP:   3000::100
Accounting Port: 1646
Authen Port:  1645
Test Username:  <Not Configured>
Test Idle Time:  60 Minutes
Test Ports:   Authen and Accounting
Server State:  Active
    Current duration 765s, previous duration 0s
  Dead: total time 0s, count 0
  Statistics:
   Authen: request 15, timeouts 1
   Author: request 0, timeouts 0
   Account: request 0, timeouts 0

Ruijie# configure terminal
Ruijie(config)# line vty 0
Ruijie(config-line)# login authentication test
Ruijie(config-line)# end
Ruijie# show running-config
!
aaa new-model
!
!
aaa authentication login test group radius
!
!
!
radius-server host 3000::100 auth-port 1645 acct-port 1646
radius-server key aaa
!
line con 0
line vty 0
login authentication test
line vty 1 4

!
```

Configuring TACACS+

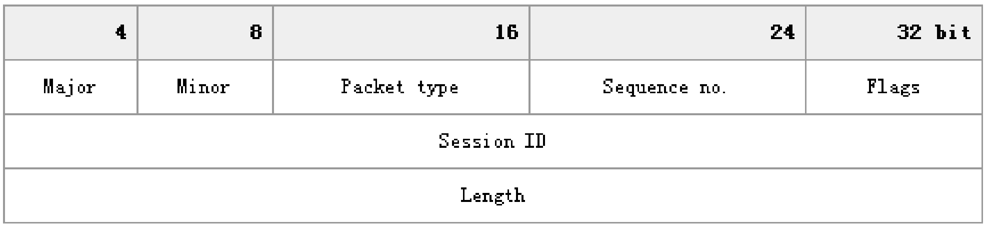
Overview of TACACS+

TACACS+ is an enhancement of Terminal Access Controller Access Control System (TACACS) defined in RFC 1492. It implements authentication, authorization, and accounting (AAA) functions on multiple types of users by communicating with the TACACS server in client-server mode. Before using the TACACS+ server, you need to configure the related functions of the TACACS+ server.

TACACS+ supports user authentication, authorization and accounting. That is, one server is used for authentication, one for authorization, and another for accounting, which proceed concurrently. Each server has its own user data for authentication, authorization, and accounting.

The following table shows the format of a TACACS+ packet:

Figure 36



4	8	16	24	32 bit
Major	Minor	Packet type	Sequence no.	Flags
Session ID				
Length				

- Major Version —TACACS+ Version;
- Minor Version —TACACS+ release;
- Packet Type — Its values are as follows:
  TAC_PLUS_AUTHEN: = 0x01 (Authentication);
  TAC_PLUS_AUTHOR:= 0x02 (Authorization);
  TAC_PLUS_ACCT:= 0x03 (Accounting).
- Sequence Number — packet sequence number in the current session. The sequence number of the first TACACS+ packet in a session must be 1 and those of subsequent packet increment by one. Therefore, the client sends only the packet with an odd sequence number, while TACACS+ Daemon only sends only the packet with an even sequence number.
- Flags — This field includes flags with various bitmap formats. The Flag value indicates whether a packet is encrypted or not.
- Session ID — ID in a TACACS+ session.
- Length —body length of a TACACS+ packet (excluding the header). All the packets are transmitted in the network after being encrypted.

TACACS+ Application

Typically, TACACS+ is used to manage and control the login of terminal users. Network devices work as TACACS+ clients to send user names and passwords to the TACACS+ server for authentication. After authentication and authorization, you can log in to the switch for operation, as shown in Figure 2:

Figure 37

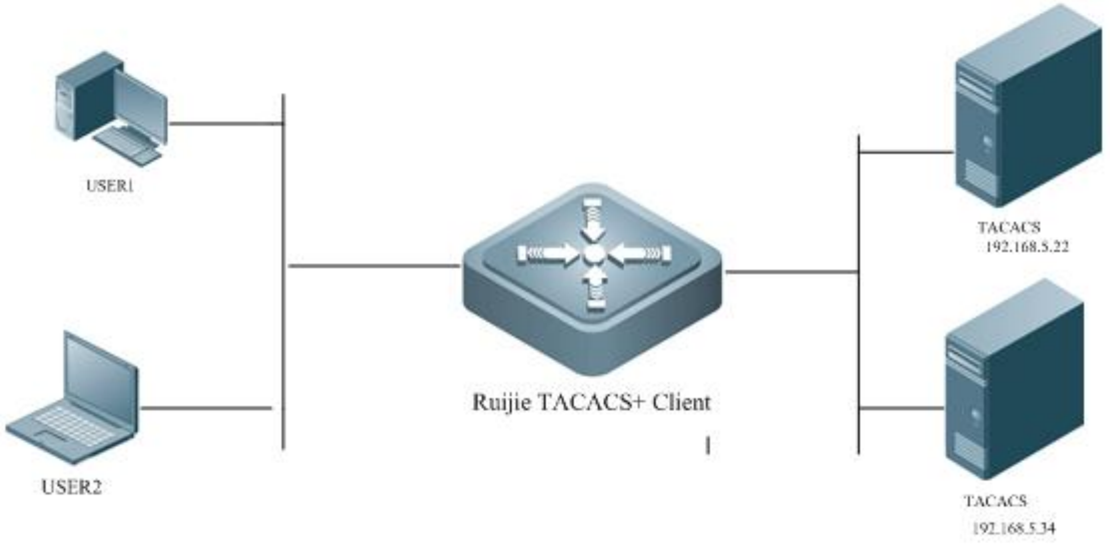
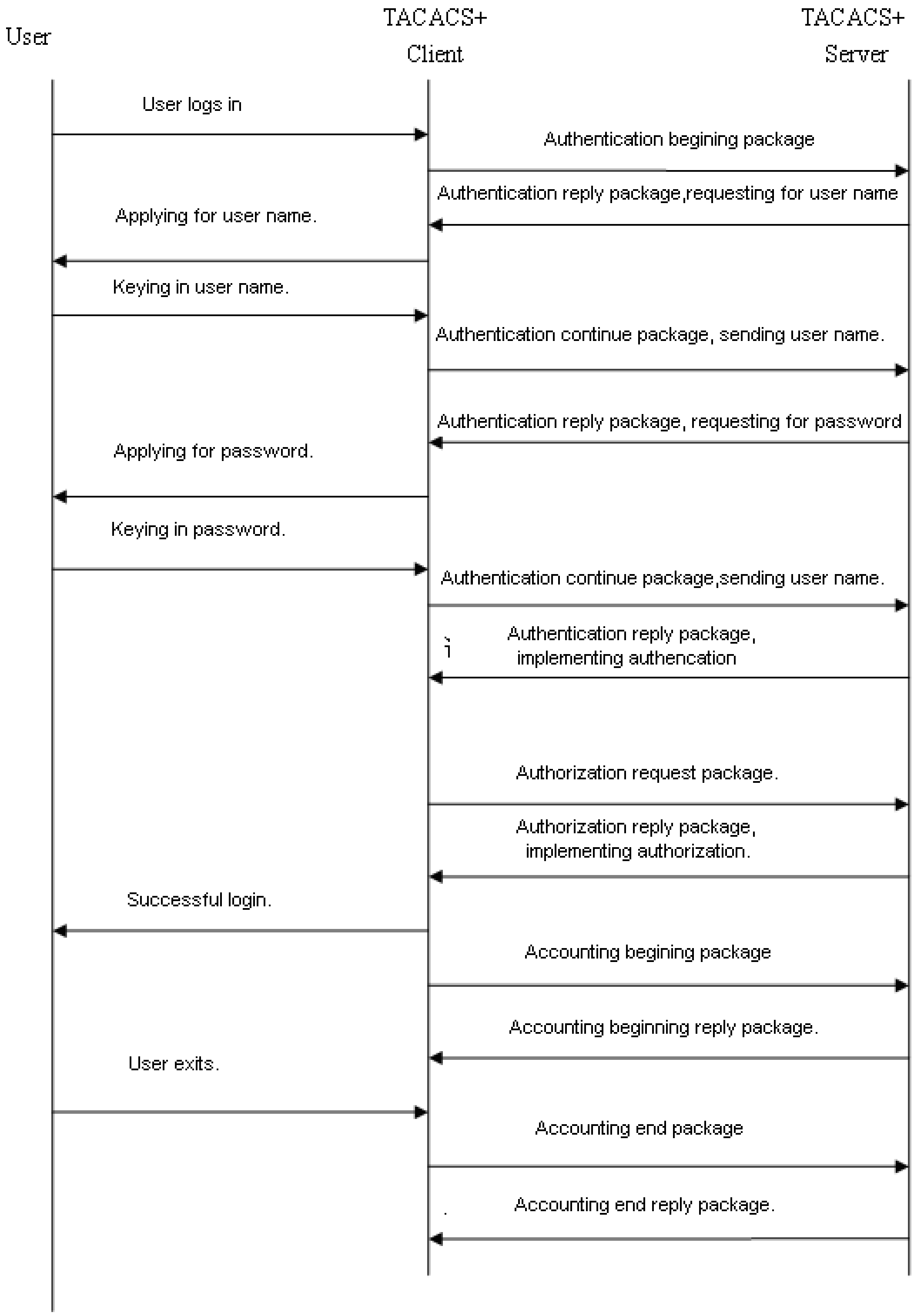


Figure describes the exchange of TACACS+ packets during AAA implementation in a login attempt.

Figure 38


User
TACACS+ Client
TACACS+ Server
User logs in
Authentication begining package
Authentication reply package,requesting for user name
Applying for user name.
Keying in user name.
Authentication continue package, sending user name.
Authentication reply package, requesting for password
Applying for password.
Keying in password.
Authentication continue package,sending user name.
Authentication reply package, implementing authencation
Authorization request package.
Authorization reply package, implementing authorization.
Successful login.
Accounting begining package
Accounting beginning reply package.
User exits.
Accounting end package
Accounting end reply package.

The whole process is divided into three parts:

Authentication:

70) A user sends a login request to the network device;

71) After receiving the request, the TACACS+ client sends a authentication start message to the TACACS+ server;

72) The TACACS+ server sends an authentication reply message, requesting the user name;
73) The TACACS+ client asks the user for the user name.
74) The user enters the login user name;
75) After receiving the user name, the TACACS+ client sends an authentication continue message containing the user name to the TACACS+ server;
76) The TACACS+ server sends an authentication reply message, requesting the login password;
77) The TACACS+ client receives the login password;
78) The user enters the login password;
79) After receiving the login password, the TACACS+ client sends an authentication continue message containing the login password to the TACACS+ server;
80) The TACACS+ server sends an authentication reply message, indicating that the user has been authenticated.

Authorization:

81) The TACACS+ client sends an authorization request message to the TACACS+ server.
82) The TACACS+ server sends an authorization reply message, indicating that the user has been authenticated;
83) The TACACS+ client receives a successful authorization reply message, displaying the interface for configuring the network device.

Accounting:

84) The TACACS+ client sends an accounting start message to the TACACS+ server;
85) The TACACS+ server sends an accounting reply message, indicating that it has received the accounting start message;
86) The user exits;
87) The TACACS+ Client sends an accounting end message to the TACACS+ server;
88) The TACACS+ server sends an accounting end reply message, indicating that it has received the accounting end message.

TACACS+ Configuration Task

The following tasks must be executed before you configure TACACS+ on the network device:

- Use the **aaa new-mode** command to enable AAA. Before using TACACS+, you must enable AAA. For usage of the **aaa new-mode** command, see the “AAA Overview” chapter in *AAA-SCG*.
- Use the **tacacs-server host** command to configure one or multiple TACACS+ servers.
- Use the **tacacs-server key** command to specify the key shared by the server and the network device.
- Use the **tacacs-server timeout** command to specify the timeout time for waiting a reply from the server;
- If authentication is required, use the **aaa authentication** command to define a TACACS+ authentication method list. For details, see the “Configuring Authentication” section in *AAA-SCG*.
- If authorization is required, use the **aaa authorization** command to define a TACACS+ authorization method list. For details, see the “Configuring Authorization” section in *AAA-SCG*.
- If accounting is required, use the **aaa accounting** command to define a TACACS+ accounting method list. For details, see the “Configuring Accounting” section in *AAA-SCG*.
- Apply a specific authentication method list to a specific line. Otherwise, a default method list is used.

Configuring TACACS+ Parameters

Before configuring TACACS+ on a network device, ensure that communication with the TACACS+ server is proper. To configure TACACS+ parameters, run the following commands:

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **tacacs-server host** {*ip-address* | *ipv6-address*} [**port** *integer*] [**timeout** *integer*] [**key** *string*]	Configures the IP address of the remote TACACS+ security server. Different combinations of parameters are used to build parameters of the server. *ip-address*: IP address of the server; *ipv6-address:* IPv6 address of the server; port *intege* [optional]:port used by the server. By default, port 49 is used. The value ranges from 1 to 65535. timeout *integer* [optional]: response timeout time of the server. By default, the timeout time is 5s. The value ranges from 1 to 1000 (in seconds). key *string* [optional]: key shared with the server with the corresponding server.
Ruijie(config)# **tacacs-server key** *string*	Configures the shared key used for communication between the network device and the TACACS+ server. When the corresponding server does not have an independent key, the global configuration is used.
Ruijie(config)# **tacacs-server timeout** *seconds*	Configures the wait time before the network device retransmits a request. It is 5s by default. If no timeout time is specified for a host, the host uses the global configuration.
Ruijie(config)# **ip tacacs source-interface** *interface*	Configures the source IP address used to send a TACACS+ request to the server. By default, the source IP address is not specified.
Ruijie(config)# **aaa group server tacacs+** *group-name*	Configures TACACS+ server groups. Different TACACS+ servers are divided into different groups.
Ruijie(config-gs-tacacs)# **server** {*ip-address* | *ipv6-address*}	Configures IP addresses of servers in a TACACS+ server group.
Ruijie(config-gs-tacacs)# **ip vrf forwarding** *vrf-name*	Configures the VRF instance name used by a TACACS+ server group. This command is available on VRF-capable hosts.

⚠ Caution

When configuring TACACS+, you must configure the TACACS+ key. The network device and the TACACS+ server must use the same shared key.

The **tacacs-server timeout** is affected by **ip tcp syntime-out,** the real valid timeout value is that of the smaller one between the two.

Using TACACS+ for Implementing AAA Functions

In a typical TACACS+ network, the TACACS+ server implements AAA functions on users. The following example shows how AAA functions are implemented through TACACS+.

Using TACACS+ for Login Authentication

- Enables AAA:

```
Ruijie# configure terminal
Ruijie(config)# aaa new-model
```

- Configures TACACS+ server information:

```
Ruijie(config)# tacacs-server host 192.168.12.219
Ruijie(config)# tacacs-server key aaa
```

- Configures TACACS+ authentication methods:

```
Ruijie(config)# aaa authentication login test group tacacs+
```

- Applies the authentication method to the interface:

```
Ruijie(config)# line vty 0 4
Ruijie (config-line)# login authentication test
```

Through the above configuration, TACACS+ login authentication is implemented. The configuration is as follows:

```
Ruijie#show running-config
!
aaa new-model
!
aaa authentication login test group tacacs+
!
tacacs-server host 192.168.12.219
tacacs-server key aaa
!
line con 0
line vty 0 4
login authentication test
!
```

Using TACACS+ for Enable Authentication

89) Enables AAA:

```
Ruijie# configure terminal
Ruijie(config)# aaa new-model
```

90) Configures TACACS+ server information:

```
Ruijie(config)# tacacs-server host 192.168.12.219
Ruijie(config)# tacacs-server host 192.168.12.218
Ruijie(config)# tacacs-server host 192.168.12.217
Ruijie(config)# tacacs-server key aaa
```

Configures that some servers in a TACACS+ server group are used:

```
Ruijie(config)# aaa group server tacacs+ tacgroup1
Ruijie(config-gs-tacacs)# server 192.168.12.219
Ruijie(config-gs-tacacs)# server 192.168.12.218
```

91) Configures to use authentication methods of TACACS+ server group 1:

```
Ruijie(config)# aaa authentication enable default group tacgroup1
```

Through the above configuration, TACACS+ Enable authentication is implemented on some servers. The configuration is as follows:

```
Ruijie#show running-config
!
aaa new-model
!
!
aaa group server tacacs+ tacgroup1
server 192.168.12.219
server 192.168.12.218
!
aaa authentication enable default group tacgroup1
!
!
tacacs-server host 192.168.12.219
tacacs-server host 192.168.12.218
tacacs-server host 192.168.12.217
tacacs-server key aaa
!
line con 0
line vty 0 4
!
```

Using TACACS+ for Login Authorization

92) Enables AAA:

```
Ruijie# configure terminal
Ruijie(config)# aaa new-model
```

93) Configures TACACS+ server information:

```
Ruijie(config)# tacacs-server host 192.168.12.219
Ruijie(config)# tacacs-server key aaa
```

94) Configures the authorization method of using tacacs+:

```
Ruijie(config)# aaa authorization exec test group tacacs+
```

95) Applies the authorization method to the interface:

```
Ruijie(config)# line vty 0 4
Ruijie (config-line)# authorization exec test
```

Through the above configuration, TACACS+ Enable authorization is implemented. The configuration is as follows:

```
Ruijie#show running-config
!
```

```
aaa new-model
!
!
aaa authorization exec test group tacacs+
!
tacacs-server host 192.168.12.219
tacacs-server key aaa
!
line con 0
line vty 0 4
authorization exec test
!
```

Using TACACS+ for Level 15 Command Audit

- Enables AAA:

```
Ruijie# configure terminal
Ruijie(config)# aaa new-model
```

- Configures TACACS+ server information:

```
Ruijie(config)# tacacs-server host 192.168.12.219
Ruijie(config)# tacacs-server key aaa
```

- Configures to use the accounting method of TACACS+:

```
Ruijie(config)# aaa accounting commands 15 test start-stop group tacacs+
```

- Applies the accounting method to the interface:

```
Ruijie(config)# line vty 0 4
Ruijie (config-line)# accounting commands 15 test
```

Through the above configuration, TACACS+ Enable accounting is implemented. The configuration is as follows:

```
Ruijie# show running-config
!
aaa new-model
!
!
aaa accounting commands 15 default group tacacs+
!
!
tacacs-server host 192.168.12.219
tacacs-server key aaa
!
line con 0
line vty 0 4
accounting commands 15 test
!
```

Configuring NAT

NAT Overview

Before Network Address Translation (NAT) configuration, it is necessary to understand the allocation of internal local addresses and internal global addresses. Perform the following configuration tasks according to different requirements.

Configuring Static NAT for Internal Source Addresses

To enable an internal network to communicate with an external network, you need to configure NAT to translate internal private IP addresses into a globally unique IP address. In this case, you can choose to configure static NAT or dynamic NAT or even both of them.

Static NAT is to establish a one-to-one permanent mapping between internal local addresses and internal global addresses. It is necessary when an external network uses a fixed global address to access hosts on an internal network. To configure static NAT, run the following commands in global configuration mode:

Command	Function
Ruijie(config)# **ip nat inside source static** *local-address global-address* [**permit-inside**] [**vrf** vrf_*name*]	Defines the static translation relationship of internal source addresses.
Ruijie(config)# **interface** *interface-type interface-number*	Enters interface configuration mode.
Ruijie(config-if)# **ip nat inside**	Defines the internal network the interface connects to.
Ruijie(config)# **interface** *interface-type interface-number*	Enters interface configuration mode.
Ruijie(config-if)# **ip nat outside**	Defines the external network the interface connects to.

The above configuration is the simplest one. You may configure several inside and outside interfaces.

Dynamic NAT is to establish a temporary mapping between internal local addresses and the internal global address pool, which will be deleted after a while. To configure dynamic NAT, run the following commands in global configuration mode:

Command	Function
Ruijie(config)# **ip nat pool** *address-pool start-address end-address* {**netmask** *mask* | **prefix-length** *prefix-length*}	Defines a global IP address pool.
Ruijie(config)# **access-list** *access-list-number* **permit** *ip-address wildcard*	Defines an ACL. Only the IP addresses that match the ACL are translated.
Ruijie(config)# **ip nat inside source list** *access-list-number* **pool** *address-pool* [**vrf** vrf_*name*]	Defines the dynamic translation relationship of internal source addresses.
Ruijie(config)# **interface** *interface-type interface-number*	Enters interface configuration mode.

Command	Function
Ruijie(config-if)# **ip nat inside**	Defines the internal network the interface connects to.
Ruijie(config)# **interface** *interface-type interface-number*	Enters interface configuration mode.
Ruijie(config-if)# **ip nat outside**	Defines the external network the interface connects to.

Note

Only source addresses that match the ACL are translated. Note that the last rule of the ACL contains a deny any statement. The ACL should not permit a wide range of IP addresses to be translated; otherwise, unexpected results will be received.

Configuring NAPT for Internal Source Addresses

Traditional NAT generally defines a one-to-one mapping and cannot enable all hosts on an internal network to communicate with an external network. NAPT allows multiple internal local addresses to be mapped to an internal global address.

NAPT is classified into static NAPT and dynamic NAPT. Static NAPT maps the designated port of a designated internal host to a designated global port, whereas static NAT maps an internal address to a global address.

To configure static NAPT, run the following commands in global configuration mode:

Command	Function
Ruijie(config)# **ip nat inside source static** {**UDP | TCP**} *local-address port global-address port* [**permit-inside**] [**vrf** vrf_*name*]	Defines the static translation relationship of internal source addresses.
Ruijie(config)# **interface** *interface-type interface-number*	Enters interface configuration mode.
Ruijie(config-if)# **ip nat inside**	Defines the internal network the interface connects to.
Ruijie(config)# **interface** *interface-type interface-number*	Enters interface configuration mode.
Ruijie(config-if)# **ip nat outside**	Defines the external network the interface connects to

Dynamic internal source address translation mentioned in previous section has automatically completed the internal source address dynamic NAPT and the configuration is to run the following command in global configuration mode.

Command	Function
Ruijie(config)# **ip nat pool** *address-pool start-address end-address* {**netmask** *mask* | **prefix-length** *prefix-length*}	Defines a global IP address pool. For NAPT, only one IP address is defined.
Ruijie(config)# **access-list** *ccess-list-number* **permit** *ip-address* **wildcard**	Defines an ACL. Only the IP addresses that match the ACL are translated.

Command	Function
Ruijie(config)# **ip nat inside source list** *access-list-number* {[**pool** *address-pool*] | [**interface** *interface-type interface-number*]} **overload** [**vrf** vrf*_name*]	Defines the dynamic translation relationship of source address. The translation effect is the same with or without overload, which is only for compatibility with mainstream manufacturers.
Ruijie(config)# **interface** *interface-type interface-number*	Enters interface configuration mode.
Ruijie(config-if)# **ip nat inside**	Defines the internal network the interface connects to.
Ruijie(config)# **interface** *interface-type interface-number*	Enters interface configuration mode.
Ruijie(config-if)# **ip nat outside**	Defines the external network the interface connects to.

NAPT may use the IP addresses in the address pool or directly uses the IP address of the interface. Generally, one address is enough to meet the address translation need of a network and can be translated into up to 64,512 addresses. In case of insufficient addresses, you can add IP addresses to the address pool.

Configuring NAT Overlap

Address Overlapping refers to the fact that two private networks in need of interconnection are allocated the same IP address or one private network and public network are allocated the same global IP address. Communication is impossible between two network hosts with overlapping addresses since they deem their counterparts are in the local network. NAT overlap is configured to solve this problem by presenting the address of external network host as that of another network host and vice versa.

NAT Overlap configuration is actually divided into two parts: 1) Internal source address translation configuration; and 2) External source address translation configuration, which is only needed by an external network that has addresses overlapped with the inner network. Static NAT or dynamic NAT may be adopted for external source address translation.

To configure static NAT for external source addresses, run the following command in global configuration mode:

Command	Function
Ruijie(config)# **ip nat outside source static** *global-address local-address* [vrf vrf*_name*]	Defines the static translation relationship of external source addresses.
Ruijie(config)# **interface** *interface-type interface-number*	Enters interface configuration mode.
Ruijie(config-if)# **ip nat inside**	Defines the internal network the interface connects to.
Ruijie(config)# **interface** *interface-type interface-number*	Enters interface configuration mode.
Ruijie(config-if)# **ip nat outside**	Defines the external network the interface connects to.

☑ NPE80 does not support NAT overlap.

Configuring TCP Load Balancing

When TCP traffic overload is detected on an internal host, more hosts can be deployed to balance the TCP traffic. In this case, you may use NAT for TCP traffic load balancing. NAT creates a virtual host, which corresponds to several real hosts, to provide TCP services, so that destination addresses are polled for load balancing. To configure destination address polling, run the following commands in global configuration mode:

Command	**Function**
Ruijie(config)# **ip nat pool** *address-pool start-address end-address* {**netmask** *mask* | **prefix-length** *prefix-length*}	Defines an IP address pool. The IP addresses of all real hosts are included in the pool.
Ruijie(config)# **access-list** *access-list-number permit ip-address* **wildcard**	Defines an ACL to match the IP address of a virtual host. The ACL should be an extended ACL used to match destination IP addresses.
Ruijie(config)# **ip nat inside destination list** *access-list-number* **pool** *address-pool* [**vrf** vrf*_name*]	Defines the dynamic translation relationship of internal destination addresses.
Ruijie(config)# **interface** *interface-type* *interface-number*	Enters interface configuration mode.
Ruijie(config-if)# **ip nat inside**	Defines the internal network the interface connects to.
Ruijie(config)# **interface** *interface-type interface-number*	Enters interface configuration mode.
Ruijie(config-if)# **ip nat outside**	Defines the external network the interface connects to.

☑ NPE80 does not support TCP load balancing.

Configuring Special Protocol Gateway

The special protocol gateways are all enabled by default. You can disable a specific special protocol gateway by running a command. All of the special protocols are just switch commands except the File Transfer Protocol (FTP) and DNS protocol, which carry parameters. Run the following commands in global configuration mode:

Command	**Function**
Ruijie(config)# **no ip nat translation ftp**	Disables the FTP gateway.
Ruijie(config)# **ip nat translation ftp**	Enables the FTP gateway. The default port is Port 21.
Ruijie(config)# **ip nat translation ftp 2121**	Enables the FTP gateway and designates Port 2121.

Configuring NAT Static Port Range Mapping

The network address translation (NAT) function can map a TCP/UDP port to an IP address one to one. However, if a continuous range of ports of the internal hosts need to be mapped to an external network, configure the NAT static port range mapping function to map the ports in the range. The port range is 1–65535.

Command	**Function**

Ruijie(config)# **ip nat inside source static** {**UDP** | **TCP**} *local-address global-address* **port-range** *min-port max-port* { [**permit**-**inside**] | [**vrf** vrf_*name*] | [**match** *interface-type interface-number*] }	Defines the static translation relationship of the internal source addresses, for mapping the IP address of an external network.
Ruijie(config)# **ip nat inside source static** {**UDP** | **TCP**} *local-address* **interface** *interface-type interface-number* **port-range** *min-port max-port* [**vrf** vrf_*name*]	Defines the static translation relationship of the internal source addresses, for mapping the interface address of an external network.
Ruijie(config)# **interface** *interface-type interface-number*	Enters the interface configuration mode.
Ruijie(config-if)# **ip nat inside**	Defines the interface for connecting to the internal network.
Ruijie(config)# **interface** *interface-type interface-number*	Enters the interface configuration mode.
Ruijie(config-if)# **ip nat outside**	Defines the interface for connecting to the external network.

Configuring ARP Response Function of NAT

In the VRRP dual-node hot backup scenario, if the same NAT function is configured on both the master and slave devices, when the router initiates an ARP request, both devices receive the ARP request and return an ARP response, causing an ARP conflict. Therefore, the slave device does not respond to the ARP request by default. If NAT configurations on the master and slave devices are different, sometimes the slave device needs to respond to the ARP request. In this case, you can run the ip nat arp reply command, so that the slave device returns a response packet when receiving an ARP request. You can run the no form of this command so that the slave device dose not return a response packet when receiving an ARP request. Run the following commands in global configuration mode:

Command	**Function**
Ruijie(config)# **ip nat arp reply**	Enables the ARP response function of NAT on the slave device.
Ruijie(config)# **no ip nat arp reply**	Disables the ARP response function of NAT on the slave device.

Configuring NAT Keepalive Function

The NAT keepalive function periodically sends ARP requests destined for the addresses in the NAT address pool to verify that the addresses exist. You can run the ip nat keepalive command to enable the keepalive function, or run the no form of this command to send no ARP request packets. The keepalive interval ranges from 1 to 86400 seconds. Run the following commands in global configuration mode:

Command	Function
Ruijie(config)# **no ip nat keepalive**	Disables the NAT keepalive function.
Ruijie(config)# **ip nat keepalive** *interval-num*	Enables the NAT keepalive function.

NAT Configuration Examples

Dynamic translation of internal source addresses

In the following configuration, local and global addresses are allocated from the NAT address pool of Net200, which defines the address range from 200.168.12.2 to 200.168.12.100. A NAT entry is created only when a packet whose internal source address matches ACL 1.

```
!
interface FastEthernet 0/0
ip address 192.168.12.1 255.255.255.0
ip nat inside
!
interface FastEthernet 1/0
ip address 200.168.12.1 255.255.255.0
ip nat outside
!
ip nat pool net200 200.168.12.2 200.168.12.100 netmask 255.255.255.0
ip nat inside source list 1 pool net200
!
access-list 1 permit 192.168.12.0 0.0.0.255
```

Reuse of internal global addresses

Reuse of internal global address is equivalent to NAPT actually. RGOS 8.1 and later versions automatically implement NAPT for dynamic NAT. In the following configuration, local and global addresses are allocated from NAT address pool—Net200, which only defines one IP address 200.168.12.200 that can be reused. A NAT entry is created only when a packet whose internal source address matches ACL 1.

```
!
interface FastEthernet 0/0
ip address 192.168.12.1 255.255.255.0
ip nat inside
```

```
!
interface FastEthernet 1/0
ip address 200.168.12.200 255.255.255.0
ip nat outside
!
ip nat pool net200 200.168.12.200 200.168.12.200 netmask 255.255.255.0
ip nat inside source list 1 pool net200
access-list 1 permit 192.168.12.0 0.0.0.255
Whether correct NAT entries can be created can be checked by looking up the NAT mapping
table.
Ruijie# show ip nat translations
Pro Inside global  Inside local  Outside local  Outside global
tcp 200.168.12.200:2063 192.168.12.65:2063 168.168.12.1:23 168.168.12.1:23
```

Static NAPT for Internal Source Addresses

Static NAPT may be used for creating a virtual server.Creating a virtual server here refers to setting up a server and mapping it to an external network through static NAPT. Thus, access to the virtual server with a global address is diverted to an internal server.

The following example describes how to map IP address 192.168.12.3 of an internal web server to a global IP address 200.198.12.1 of port 80. The configuration script is as follows:

```
!
interface FastEthernet 0/0
ip address 192.168.12.1 255.255.255.0
ip nat inside
!
interface FastEthernet 1/0
ip address 200.198.12.1 255.255.255.0
ip nat outside
!
ip nat inside source static tcp 192.168.12.3 80 200.198.12.1 80
```

For details, see the “Configuring a local server” section.

TCP Load Balancing

A virtual host address is defined in the following configuration so that all TCP connections to this virtual host from external networks will be processed by multiple real hosts for load balancing. **Realhosts** defines a real host address pool, while ACL 1 defines the IP address of the virtual host. Traffic from hosts on an external network must be routed to this virtual host. The following configuration applies only to TCP traffic. Note that an extended ACL must be configured to match destination IP addresses.

```
!
interface FastEthernet 0/0
ip address 10.10.10.1 255.255.255.0
ip nat inside
```

```
!
interface FastEthernet 1/0
ip address 200.198.12.1 255.255.255.0
ip nat outside
!
ip nat pool realhosts 10.10.10.2 10.10.10.3 netmask 255.255.255.0 type rotary
ip nat inside destination list 100 pool realhosts
!
access-list 100 permit ip any host 10.10.10.100
!
```

Whether correct NAT entries can be created can be checked by looking up the NAT mapping table.

```
Ruijie# sh ip nat translations
Pro Inside global Inside local Outside local   Outside global
tcp 10.10.10.100:23 10.10.10.2:23 100.100.100.100:1178 100.100.100.100:1178
tcp 10.10.10.100:23 10.10.10.3:23 200.200.200.200:1024 200.200.200.200:1024
```

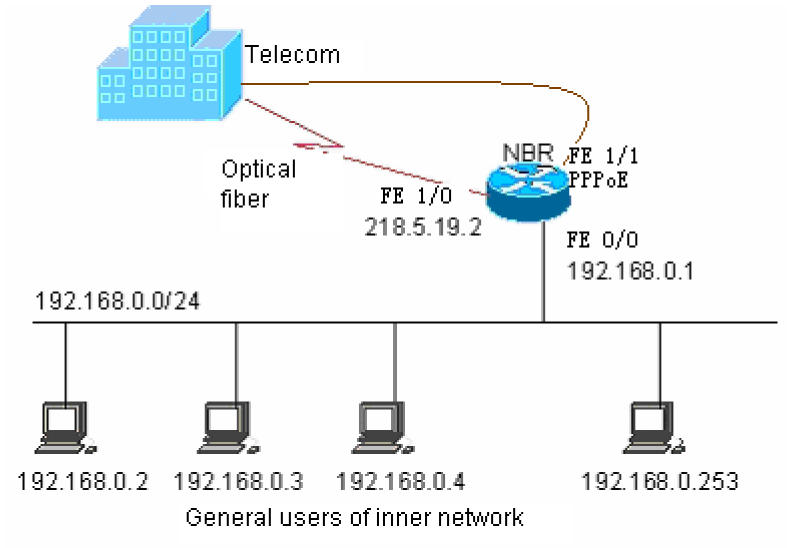
Load balancing among multiple outside interfaces

If several WAN ports of a device are used as outside interfaces, load is balanced among these WAN ports by bandwidth. When one WAN port is faulty, the load will be automatically routed to other normal ports. By default, the load is distributed according to global destination addresses of NAT. In the following example, load is balanced between two WAN ports of a RSR series router.

96) Interface GigabitEthernet 0/0 connects to a telecom network.
97) Interface GigabitEthernet 0/1 connects to the education network.

The topology is as follows:

Figure 39



The configuration is as follows:

```
!
```

# Configure an ACL to allow internal network users to access internet.

```
access-list 99 permit 192.168.0.0 0.0.0.255
```

# Configure GigabitEthernet 0/2 to connect to the internal network..

```
interface GigabitEthernet 0/2
ip nat inside
ip address 10.29.0.253 255.255.255.0
!
```

# Configure a static IP address for WAN port 0 which connects to the telecom network.

```
interface GigabitEthernet 0/0
ip nat outside
ip address 218.4.53.238 255.255.255.0
!
```

# WAN port 1 connects to the education network.

```
interface GigabitEthernet 0/1
ip nat outside
ip address 172.16.253.18 255.255.255.0
!
```

# Configure a NAT address pool. NAT provides multiple Outside ports. If GigabitEthernet 0/0 is configured as the Outside port, the IP address of the port is set to 218.4.53.238; if GigabitEthernet 0/1 is configured as the Outside port, the IP address of the port is set to 172.16.253.18.

```
ip nat pool setup_build_pool prefix-length 24
address 61.155.18.17 61.155.18.18 match interface GigabitEthernet 0/0
address 210.28.160.100 210.28.160.110 match interface GigabitEthernet 0/1
```

# Enable internal source address translation of NAT

```
ip nat inside source list 99 pool nbr_setup_build_pool
```

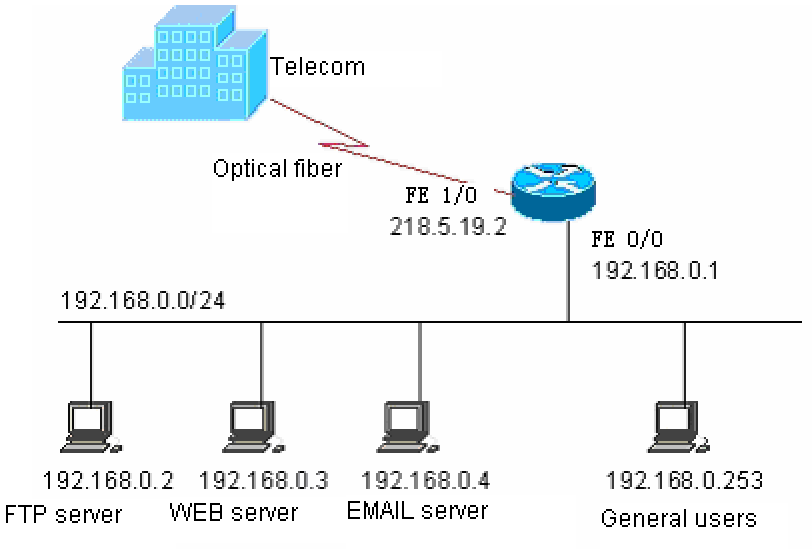
# Configure that traffic is routed to two WAN ports by default.

```
ip route 0.0.0.0 0.0.0.0 FastEthernet 1/0 202.101.98.1
ip route 0.0.0.0 0.0.0.0 dialer 1
!
```

Configuring a local server

To configure a local server means to map one or more hosts to a network access server (NAS), so that users on the WAN can access desired services. As shown in Figure , three servers (an FTP server, a web server, and an E-mail server) are deployed on the internal network. It is expected that hosts on the WAN can access the three servers and common users of the internal network can access Internet by using the gateway as a NAS. For Ruijie products, static NAT is used for server access and dynamic NAT is used for Internet access.

Figure 40 Configuring a local server



To realize these functions, static NAT needs to be configured.

# Enter privileged user mode

```
Ruijie> enable
```

# Enter global configuration mode

```
Ruijie# config terminal
Enter configuration commands, one per line. End with CNTL/Z.
```

# Enter WAN port 0 configuration mode

```
Ruijie(config)#interface fastethernet 1/0
```

# Configure the IP address of the WAN port

```
Ruijie(config-if)# ip address 218.5.19.2 255.255.255.0
```

# Configure the WAN port as the connection-sharing Internet access port

```
Ruijie(config-if)# ip nat outside
```

# Enable the WAN port

```
Ruijie(config-if)# no shut
```

# Return to common user mode

```
Ruijie(config-if)# end
Ruijie#
```

# The system prompts that the link to the WAN port is Up.

```
%LINK CHANGED: Interface FastEthernet 1/0, changed state to up
%LINE PROTOCOL CHANGE: Interface FastEthernet 1/0, changed state to UP
Ruijie# config terminal
Enter configuration commands, one per line. End with CNTL/Z.
```

# Enter LAN port configuration mode

```
Ruijie(config)# interface fastethernet 0/0
```

# Configure the IP address of the LAN port

```
Ruijie(config-if)# ip address 192.168.0.1 255.255.255.0
```

# Configure the LAN port as the connection-sharing internet access port

```
Ruijie(config-if)# ip nat inside
```

# Enable the LAN port

```
Ruijie(config-if)# no shut
Ruijie(config-if)# end
Ruijie#
%LINK CHANGED: Interface FastEthernet 0/0, changed state to up
%LINE PROTOCOL CHANGE: Interface FastEthernet 0/0, changed state to UP
Ruijie# config terminal
Enter configuration commands, one per line. End with CNTL/Z.
```

# Configure default route to access to internet

```
Ruijie(config)# ip route 0.0.0.0 0.0.0.0 fastethernet 1/0 218.5.19.1
```

# Configure a default route for Internet access

```
Ruijie(config)# ip route 0.0.0.0 0.0.0.0 fastethernet 1/0 218.5.19.1
```

# Configure an ACL for NAT application

```
Ruijie(config)# access-list 1 permit any
```

# Configure a connection sharing rule to allow common internal users to access Internet over a device

```
Ruijie(config)#ip nat inside source list 1 interface fastethernet 1/0
```

# Configure static mapping of the FTP server

```
Ruijie(config)# ip nat inside source static tcp 192.168.0.2 20 218.5.19.2 20
Ruijie(config)# ip nat inside source static tcp 192.168.0.2 21 218.5.19.2 21
```

# Configure static mapping of the web server

```
Ruijie(config)# ip nat inside source static tcp 192.168.0.3 80 218.5.19.2 80
```

# Configure static mapping of the E-mail server

```
Ruijie(config)# ip nat inside source static tcp 192.168.0.4 25 218.5.19.2 25
Ruijie(config)# ip nat inside source static tcp 192.168.0.4 110 218.5.19.2 110
Ruijie(config)# end
Ruijie#
Ruijie# config terminal
Enter configuration commands, one per line. End with CNTL/Z.
```

# Configure a password for Telnet access

```
Ruijie(config)# line vty 0 4
Ruijie(config-line)# password remoteuser
Ruijie(config-line)# end
Ruijie#
Ruijie# config terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)# enable secret private
```

# Configure a device name

```
Ruijie(config)# host RUIJIE
RUIJIE(config)# end
RUIJIE#
```

# Save the configuration

```
RUIJIE# write
Building configuration...
[OK]
RUIJIE#
```

# Verify the configuration

```
RUIJIE# show running-config
Building configuration...
Current configuration:
!
!
hostname NBR
!
!
!
access-list 1 permit any
!
!
interface FastEthernet 0/0
ip address 192.168.0.1 255.255.255.0
ip nat inside
!
interface FastEthernet 1/0
ip address 218.5.19.2 255.255.255.0
ip nat outside
!
ip nat inside source list 1 interface FastEthernet 1/0
ip nat inside source static tcp 192.168.0.4 110 218.5.19.2 110
ip nat inside source static tcp 192.168.0.4 25 218.5.19.2 25
ip nat inside source static tcp 192.168.0.3 80 218.5.19.2 80
ip nat inside source static tcp 192.168.0.2 21 218.5.19.2 21
ip nat inside source static tcp 192.168.0.2 20 218.5.19.2 20
!
ip route 0.0.0.0 0.0.0.0 FastEthernet 1/0 218.5.19.1
!
line con 0
line vty 0 4
password remoteuser
login
!
end
RUIJIE#
```

NAT Configuration in case of multiple VRF instances

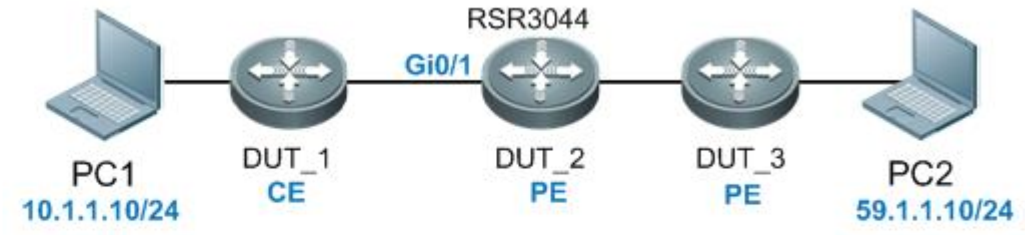
The following example shows the NAT implementation when there are multiple VRF instances. An IP address may be found in different VRF instances and needs to be translated into different source IP addresses during NAT. In this case, you must specify the target VRF domain of NAT.

```
access-list 1 permit 192.168.12.0 0.0.0.255

ip vrf 1

ip vrf 2

interface FastEthernet 0/0
ip vrf forward 1
ip address 192.168.12.1 255.255.255.0
ip nat inside
!
interface FastEthernet 0/1
ip vrf forward 1
ip address 100.168.12.200 255.255.255.0
ip nat outside
!
interface FastEthernet 1/0
ip vrf forward 2
ip address 192.168.12.1 255.255.255.0
ip nat inside
!
interface FastEthernet 1/1
ip vrf forward 2
ip address 200.168.12.200 255.255.255.0
ip nat outside
!
ip nat pool net100 100.168.12.200 100.168.12.200 netmask 255.255.255.0
ip nat pool net200 200.168.12.200 200.168.12.200 netmask 255.255.255.0

ip nat inside source list 1 pool net100 vrf 1
ip nat inside source list 1 pool net200 vrf 2
Whether correct NAT entries can be created can be checked by looking up the NAT mapping
table.
Ruijie# show ip nat translations vrf 1
Pro Inside global  Inside local  Outside local  Outside global
tcp 100.168.12.200:2063 192.168.12.65:2063 168.168.12.1:23 168.168.12.1:23
Ruijie# show ip nat translations vrf 2
Pro Inside global  Inside local  Outside local  Outside global
tcp 200.168.12.200:2063 192.168.12.65:2063 168.168.12.1:23 168.168.12.1:23
```

VPN NAT configuration example

On a MPLS network, NAT can be used to implement VRF traversal.

Figure 41



Configure MPLS

```
mpls ip
```

Configure PBR

```
route-map vrfdata permit 10
 match ip address 150
 set vrf data
```

Specify an ACL

```
ip access-list extended 100
 10 permit ip 10.0.0.0 0.255.255.255 any
ip access-list extended 150
 10 permit ip any 20.1.1.0 0.0.0.255
```

Configure VRF domains

```
ip vrf data
 rd 200:1
 route-target both 200:1
ip vrf v1
 rd 100:1
 route-target export 100:1
ip vrf v2
 rd 100:2
 route-target export 100:2
```

Deploy MPLS on a public network interface

```
interface GigabitEthernet 0/0
 ip nat outside
 ip ref
 ip address 10.3.1.3 255.255.255.0
 label-switching
 mpls ip
 duplex auto
 speed auto
```

Configure PBR on a private network interface for NAT deployment

```
interface GigabitEthernet 0/1
 ip vrf forwarding v1
 ip nat inside
 ip policy route-map vrfdata
 ip ref
 ip address 10.1.1.1 255.255.255.0
 duplex auto
 speed auto
interface GigabitEthernet 0/1.1
 encapsulation dot1Q 100
 ip vrf forwarding v2
 ip nat inside
 ip policy route-map vrfdata
 ip address 10.1.1.1 255.255.255.0
```

Configure the loopback interface for advertising routes

```
interface Loopback 0
 ip ref
 ip address 3.3.3.3 255.255.255.255
router bgp 100
 bgp router-id 3.3.3.3
 bgp log-neighbor-changes
 neighbor 4.4.4.4 remote-as 100
 neighbor 4.4.4.4 update-source Loopback 0
address-family ipv4
 neighbor 4.4.4.4 activate
 exit-address-family
address-family vpnv4 unicast
 neighbor 4.4.4.4 activate
 neighbor 4.4.4.4 send-community both
 neighbor 4.4.4.4 route-map hzb out
 exit-address-family
address-family ipv4 vrf data
 maximum-prefix 10000
 network 0.0.0.0
 redistribute connected
 redistribute static
 exit-address-family
address-family ipv4 vrf v1
 maximum-prefix 10000
 redistribute static
 exit-address-family
address-family ipv4 vrf v2
 maximum-prefix 10000
 exit-address-family
router ospf 1
 router-id 3.3.3.3
 network 0.0.0.0 255.255.255.255 area 0
mpls router ldp
 ldp router-id interface Loopback 0 force
```

Configure NAT to take effect on VRF data packets

```
ip nat pool abc 100.1.1.1 100.1.1.1 netmask 255.255.255.0
ip nat inside source static 10.1.1.2 100.1.1.1 vrf data
```

Specify a blackhole route

```
ip route vrf data 100.1.1.1 255.255.255.255 Null 0
```

NAT Static Port Range Mapping

```
interface GigabitEthernet 0/0
 ip nat inside
 ip address 11.1.1.1 255.255.255.0
 duplex auto
 speed auto
!
interface GigabitEthernet 0/1
 ip nat outside
 ip address 22.2.2.1 255.255.255.0
 duplex auto
 speed auto
!
ip nat inside source static tcp 11.1.1.2 22.2.2.3 port-range 80 100
```

You can check whether the translation record is established correctly by displaying the NAT mapping table.

```
RSR10X#show ip nat translations
Pro Inside global     Inside local      Outside local     Outside global
tcp 22.2.2.3:93       11.1.1.2:93       33.3.3.18:1000    33.3.3.18:1000
tcp 22.2.2.3:82       11.1.1.2:82       33.3.3.7:1005     33.3.3.7:1005
tcp 22.2.2.3:83       11.1.1.2:83       33.3.3.8:1006     33.3.3.8:1006
tcp 22.2.2.3:99       11.1.1.2:99       33.3.3.24:1000    33.3.3.24:1000
tcp 22.2.2.3:84       11.1.1.2:84       33.3.3.9:1007     33.3.3.9:1007
tcp 22.2.2.3:98       11.1.1.2:98       33.3.3.23:1000    33.3.3.23:1000
tcp 22.2.2.3:94       11.1.1.2:94       33.3.3.19:1000    33.3.3.19:1000
tcp 22.2.2.3:87       11.1.1.2:87       33.3.3.12:1010    33.3.3.12:1010
tcp 22.2.2.3:95       11.1.1.2:95       33.3.3.20:1000    33.3.3.20:1000
tcp 22.2.2.3:86       11.1.1.2:86       33.3.3.11:1009    33.3.3.11:1009
tcp 22.2.2.3:81       11.1.1.2:81       33.3.3.6:1004     33.3.3.6:1004
tcp 22.2.2.3:89       11.1.1.2:89       33.3.3.14:1000    33.3.3.14:1000
tcp 22.2.2.3:97       11.1.1.2:97       33.3.3.22:1000    33.3.3.22:1000
tcp 22.2.2.3:91       11.1.1.2:91       33.3.3.16:1000    33.3.3.16:1000
tcp 22.2.2.3:80       11.1.1.2:80       33.3.3.5:1003     33.3.3.5:1003
tcp 22.2.2.3:96       11.1.1.2:96       33.3.3.21:1000    33.3.3.21:1000
tcp 22.2.2.3:85       11.1.1.2:85       33.3.3.10:1008    33.3.3.10:1008
tcp 22.2.2.3:90       11.1.1.2:90       33.3.3.15:1000    33.3.3.15:1000
tcp 22.2.2.3:92       11.1.1.2:92       33.3.3.17:1000    33.3.3.17:1000
tcp 22.2.2.3:88       11.1.1.2:88       33.3.3.13:1000    33.3.3.13:1000
tcp 22.2.2.3:100      11.1.1.2:100      33.3.3.25:1000    33.3.3.25:1000
```

Configuring SSH Terminal Service

Overview of SSH

A Secure Shell (SSH) connection functions like a Telnet connection, except that all transmissions based on the connection are encrypted. When a user logs in to the device from an insecure network, the SSH feature provides information security guarantee and powerful authentication function to protect the devices from IP spoofing, plain password interception and other kinds of attacks.

Ruijie SSH service supports both the IPv4 and IPv6 protocols.

SSH Algorithms Supported by Ruijie Products

Supported Algorithm	SSH1	SSH2
Signature authentication algorithm	RSA	RSA, DSA
Key exchange algorithm	RSA public key encryption based key exchange algorithm	KEX_DH_GEX_SHA1 KEX_DH_GRP1_SHA1 KEX_DH_GRP14_SHA1
Encryption algorithm	DES, 3DES, Blowfish	DES, 3DES, AES-128, AES-192, AES-256
User authentication algorithm	User password based authentication method	User password based authentication method
Message authentication algorithm	Not supported	MD5, SHA1, SHA1-96, MD5-96
Compression algorithm	NONE	NONE

SSH Configuration

Default SSH Configurations

Item	Default Value
SSH server status	Off
SSH version	Compatible mode (supporting versions 1 and 2)
SSH user authentication timeout period	120s
SSH user re-authentication times	3

Configuring User Authentication

- Considering the SSH connection security, the login without authentication is forbidden. Therefore, in the login authentication of the users, the login authentication mode must have password configured (authentication-free login allowed for telnet).

- The username and password entered every time must be set. If the current authentication mode does not need the username, the username can be entered randomly but the length must be greater than zero.

Enabling SSH Server

The SSH Server is disabled by default. To enable the SSH Server, run the **enable service ssh-server** command in global configuration mode while generating a SSH key.

Command	Description
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **enable service ssh-server**	Enables the SSH Server.
Ruijie(config)# **crypto key generate** {**rsa|dsa**}	Generates a key

Caution

To delete a key, use the crypto key zeroize command rather than the [no] crypto key generate command. The SSH module does not support hot standby. For products supporting management module hot standby, after the management module is switched over, if the current primary board has no SSH key file, the crypto key generate command must be used to regenerate a key in order to use SSH.

Shutting Down the SSH Server

To disable the SSH Server, run the **no enable service ssh-server** command in global configuration mode:

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **no enable service ssh-server**	Disables the SSH Server.

Configuring the Supported SSH Server Version

By default, the SSHv1 and SSHv2 are compatible. Run the following commands to configure the SSH version.

Command	Function
Ruijie# **configure terminal**	Enters configuration mode.
Ruijie(config)# **ip ssh version** {**1 | 2**}	Configures the supported SSH version.
Ruijie(config)# **no ip ssh version**	Restore the default SSH version. By default, SSHv1 and SSHv2 are supported.

Configuring SSH User Authentication Timeout Period

By default, the user authentication timeout period of the SSH server is 120 seconds. Run the following commands to configure the SSH user authentication timeout period.

Command	Function
Ruijie# **configure terminal**	Enters configuration mode.
Ruijie(config)# **ip ssh time-out** *time*	Configures the SSH timeout period (ranging from 1 to 120 seconds).
Ruijie(config)# **no ip ssh time-out**	Restores the default SSH user authentication timeout period to 120 seconds.

Configuring SSH Re-authentication Times

This command is used to set the authentication attempts for SSH users requesting connections to prevent illegal actions such as malicious guesswork. By default, three authentication attempts can be made for the SSH Server. In other words, it allows the user to enter the username and password for three times to attempt the authentication. Run the following commands to configure the SSH re-authentication times:

Command	**Function**
Ruijie# **configure terminal**	Enters configuration mode.
Ruijie(config)# **ip ssh authentication-retries** *retry times*	Configures SSH re-authentication times (ranging from 0 to 5).
Ruijie(config)# **no ip ssh authentication-retries**	Restores the default SSH re-authentication times to 3.

Caution For details of the preceding commands, see SSH Command Reference Manual.

Configuring SSH Public Key Based Authentication

Only the version 2 of the SSH protocol supports public key based authentication. The following commands associate a public key file with a username. When processing client authentication, the server uses a specified public key file according to the username of the client.

Command	**Function**
Ruijie# configure terminal	Enters configuration mode.
Ruijie(config)# **ip ssh peer** *test* **public-key rsa flash**:*rsa.pub*	Configures the RSA public key file that is associated with the username *test*.
Ruijie(config)# ip ssh peer *test* **public-key dsa flash**:*dsa.pub*	Configures the DSA public key file that is associated with the username *test*.

Configuring the SCP Server Function

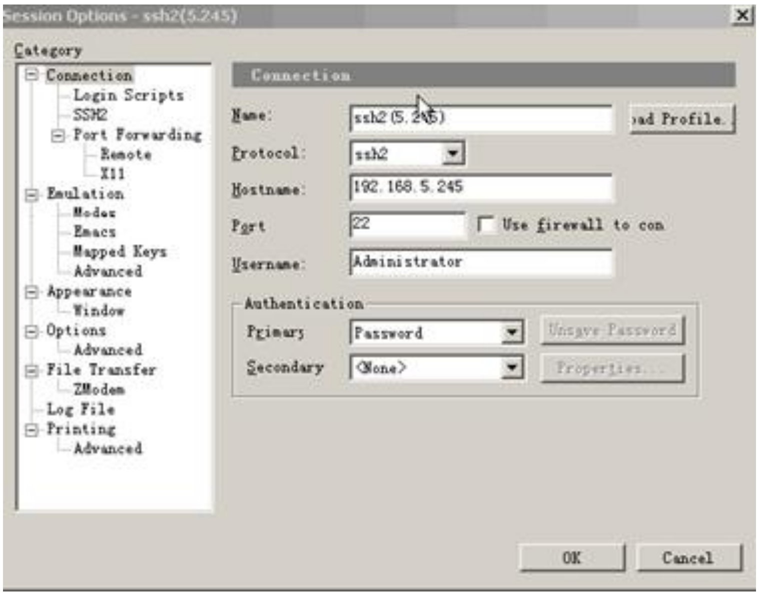
With the SCP server enabled on a network device, the user can directly download files from the network device and upload local files to the network device. Meanwhile, the user can transfer all interactive data in encrypted text manner, featuring authentication and security.

Command	Function
Ruijie# **configure terminal**	Enters configuration mode..
Ruijie(config)#**ip scp server enable**	Enable the SCP server function.
Ruijie(config)# **no ip scp server enable**	Disable the SCP server function.

Note For details of the above commands, see *SSH Command Reference Manual*.

Using SSH for Device Management

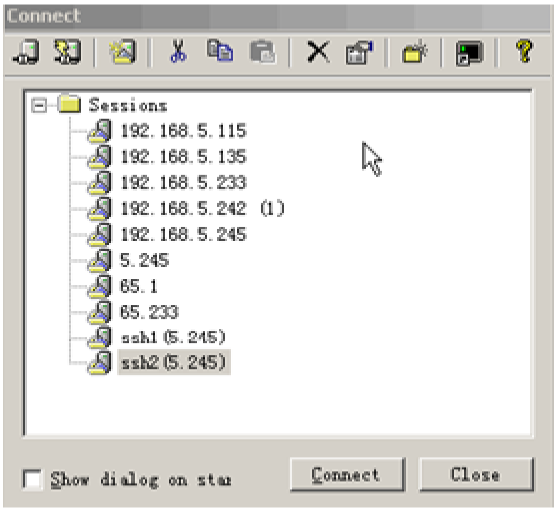
You may use SSH for device management after enabling the SSH Server function that is disabled by default. Since Telnet that comes with the Windows does not support SSH, third-party client software must be used. Currently, the clients with sound forward compatibility include Putty, Linux and SecureCRT. With the client software SecureCRT as an example, the SSH client configuration is described as follows (see the UI below):



Name: ssh2 (5.245)
Protocol: ssh2
Hostname: 192.168.5.245
Port 22 □ Use firewall to con
Username: Administrator
Primary Password
Secondary <None>

As shown in the figure, protocol 2 is used for login, so **SSH2** is selected for **Protocol**. **Hostname** indicates the IP address of the host the user will log in, 192.168.5.245. Port 22 is the default port listened by SSH. **Username** indicates the username, and does not take effect when the device only requires a password. **Authentication** indicates the authentication mode, and the username/password authentication is supported here. The used password is the same as the password used for Telnet.

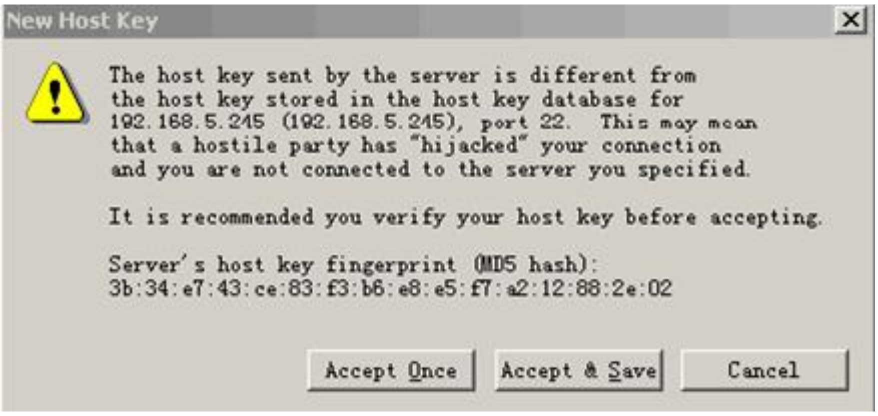
Click **OK**. The following dialog box pops up:



Sessions
192.168.5.115
192.168.5.135
192.168.5.233
192.168.5.242 (1)
192.168.5.245
5.245
65.1
65.233
ssh1 (5.245)
ssh2 (5.245)

□ Show dialog on star

Click **Connect** to log in to the host, as shown below:

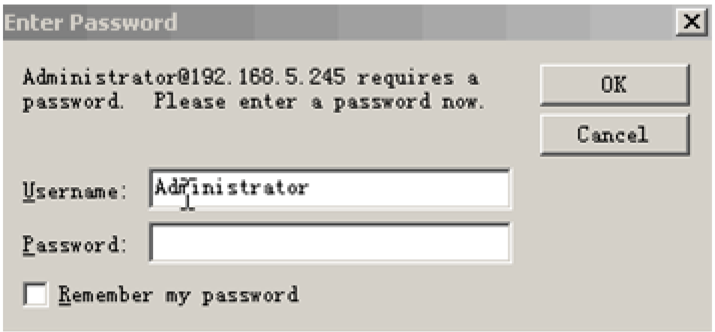


The host key sent by the server is different from the host key stored in the host key database for 192.168.5.245 (192.168.5.245), port 22. This may mean that a hostile party has "hijacked" your connection and you are not connected to the server you specified.

It is recommended you verify your host key before accepting.

Server's host key fingerprint (MD5 hash):
3b:34:e7:43:ce:83:f3:b6:e8:e5:f7:a2:12:88:2e:02

Ask the user whose is logging in to the host 192.168.5.245 whether to receive the key from the server. Select **Accept & Save** or **Accept Once**. A dialog box, prompting you to enter a password, pops up as follows:



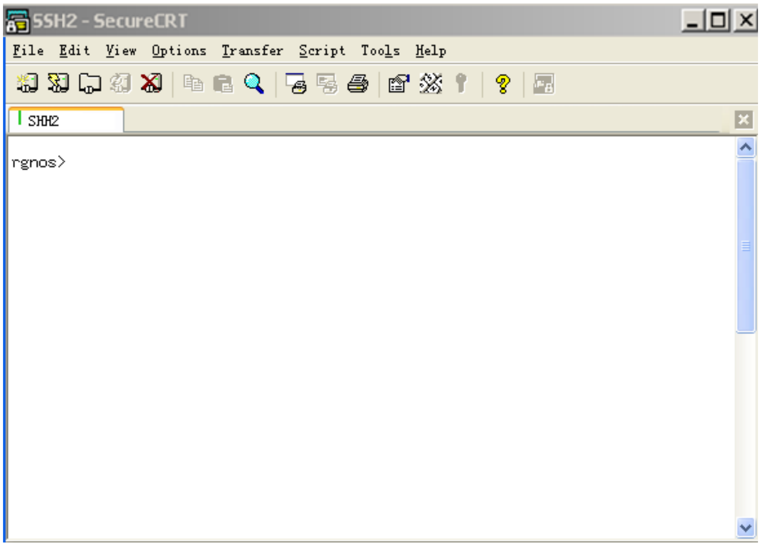
Administrator@192.168.5.245 requires a password. Please enter a password now.

Username: Administrator

Password:

□ Remember my password

Enter the Telnet login password. A window pops up as follows:



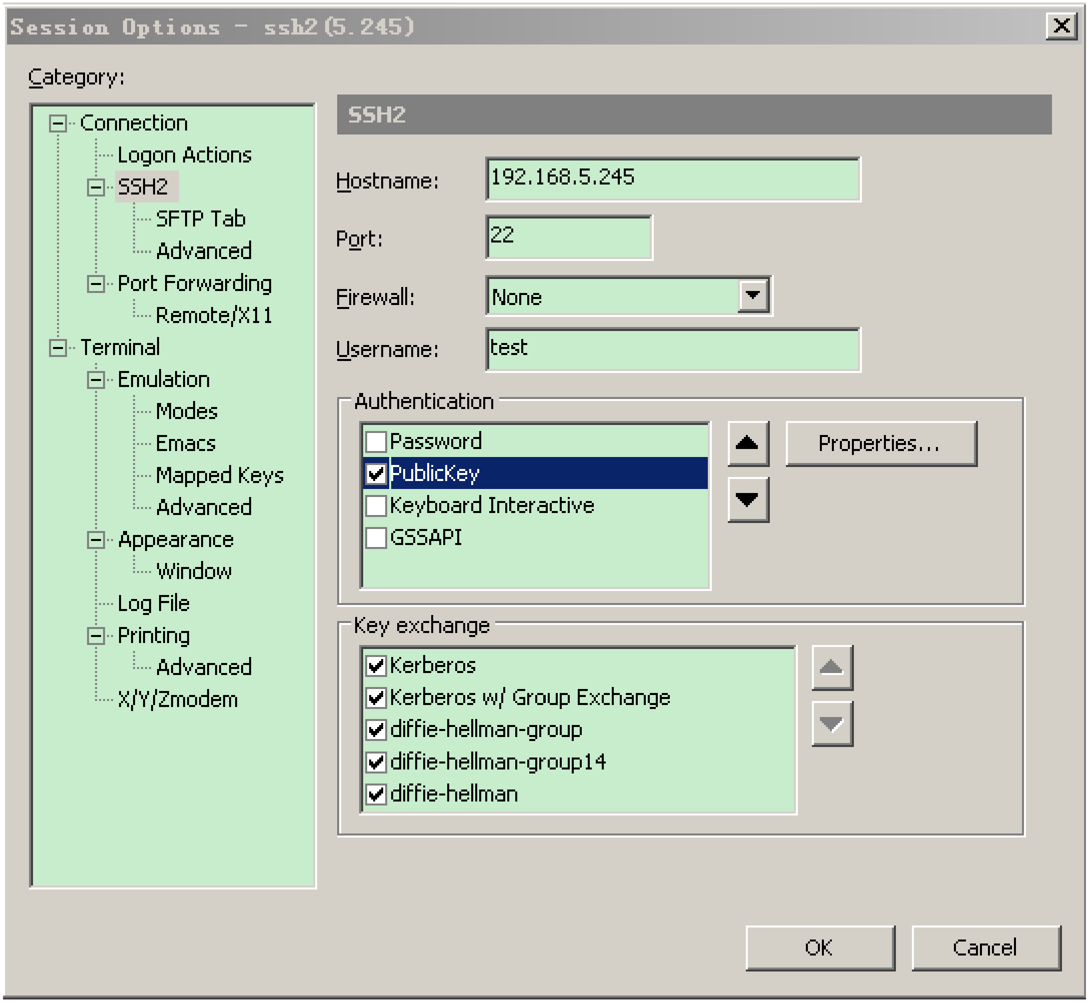
```
rgnos>
```

Enabling SSH public key based authentication

Operations on the SSH Client:

To enable SSH public key authentication, generate a key pair (RSA or DSA) on the SSH client and put the public key on the SSH server. And then enabling SSH public key based authentication on SSH server. The following section takes client software SecureCRT for an instance to demonstrate how to generate key pair on SSH client.

Firstly, click PublicKey in Authentication on Session Option, and then click Properties. as Shown below.


Session Options - ssh2(5.245)

SSH2

Hostname: 192.168.5.245

Port: 22

Firewall: None

Username: test

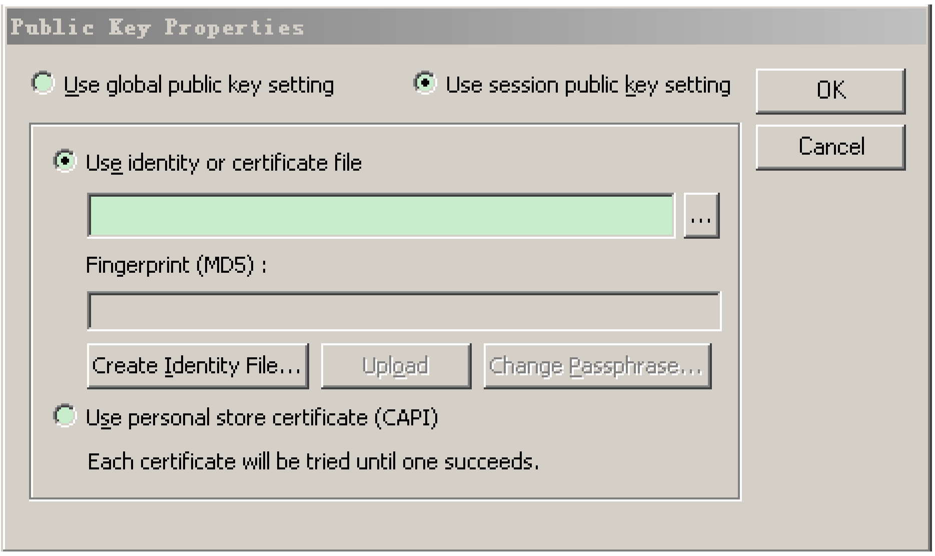
Authentication

- □ Password
- ☑ PublicKey
- □ Keyboard Interactive
- □ GSSAPI

Key exchange

- ☑ Kerberos
- ☑ Kerberos w/ Group Exchange
- ☑ diffie-hellman-group
- ☑ diffie-hellman-group14
- ☑ diffie-hellman

If the key pair is generated before, use the private key （Use identity or certificate file）. Note that this private key must be paired with the public key on the SSH server, otherwise the authentication fails. As shown below.


Public Key Properties

○ Use global public key setting ● Use session public key setting

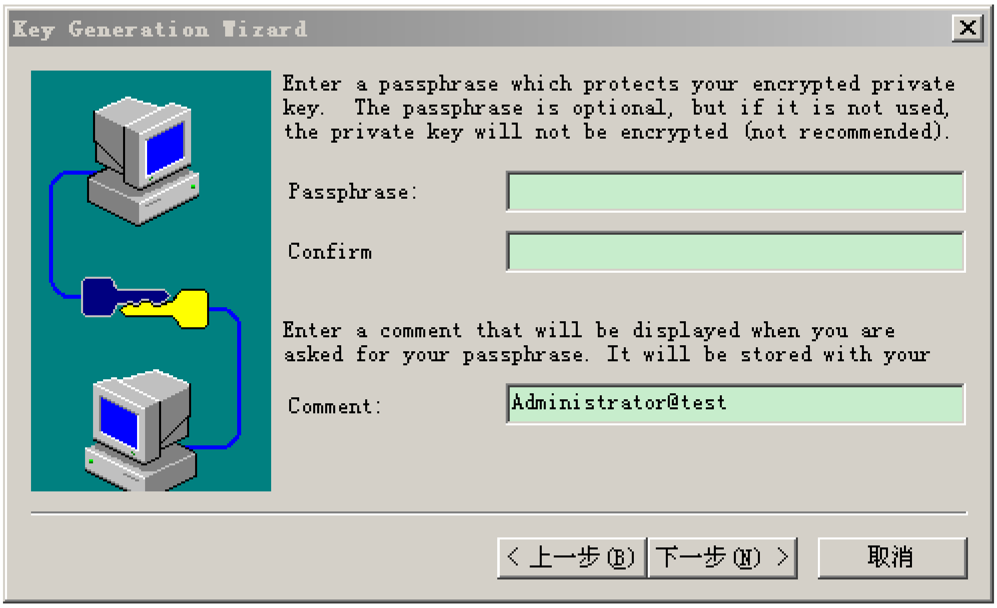
● Use identity or certificate file

Fingerprint (MD5) :

○ Use personal store certificate (CAPI)

Each certificate will be tried until one succeeds.

If no key pair is generated before, generate a new key pair. (Optional) set a passphrase for the private key. Once the passphrase is set, you should key in this passphrase for authentication. As shown below.



Enter a passphrase which protects your encrypted private key. The passphrase is optional, but if it is not used, the private key will not be encrypted (not recommended).

Passphrase:

Confirm

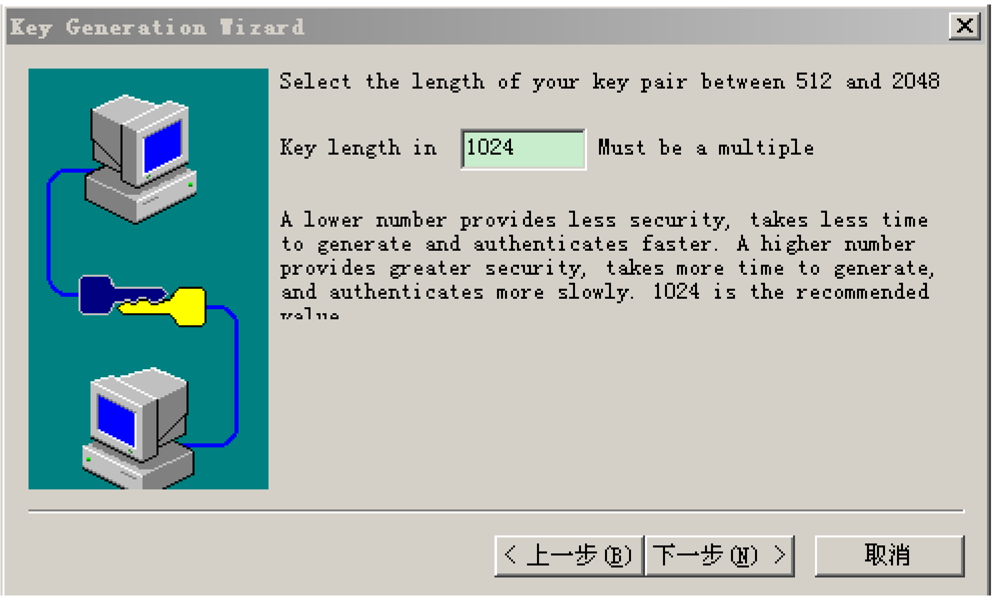
Enter a comment that will be displayed when you are asked for your passphrase. It will be stored with your

Comment: Administrator@test

Note

Shaking the mouse when the SSH client is generating the key pair, otherwise the generating rate will be slow.

The key files must be stored in OpenSSH key format, otherwise the files cannot be used. If the Putty is adopted as the client software, still the private key is required to turn into the Putty format by puttygen.exe.(Puttygen.exe can generate key pair in OpenSHH format, but Putty cannot use key pair in OpenSHH format directly). However, the public key files generated in OpenSSH format do not need transformation.



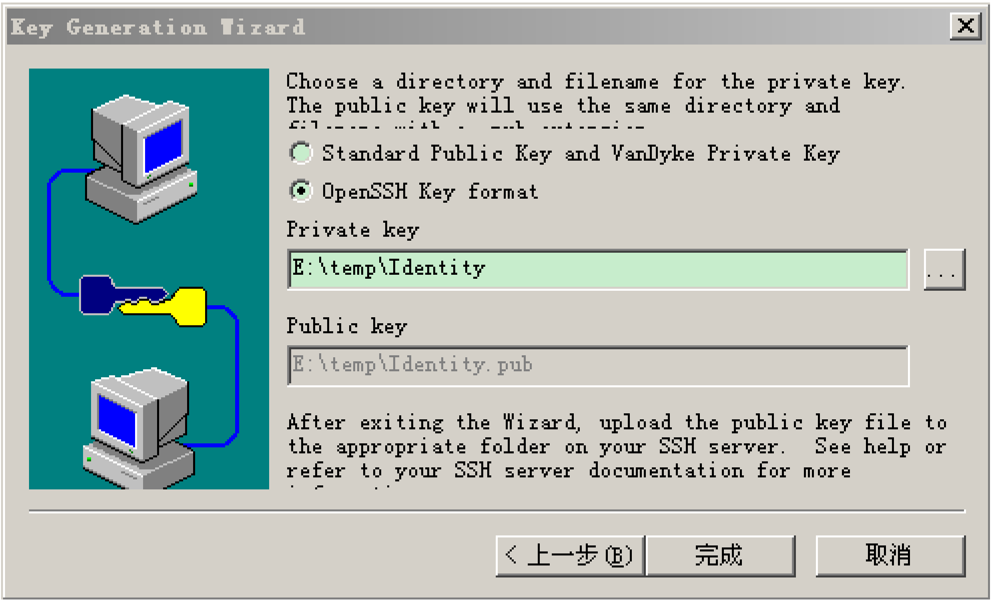
Select the length of your key pair between 512 and 2048

Key length in 1024 Must be a multiple

A lower number provides less security, takes less time to generate and authenticates faster. A higher number provides greater security, takes more time to generate, and authenticates more slowly. 1024 is the recommended value

Note

To guarantee the security of RSA public key based authentication, make sure the key length is longer or equivalent to 768.



Choose a directory and filename for the private key. The public key will use the same directory and

○ Standard Public Key and VanDyke Private Key

● OpenSSH Key format

Private key

E:\temp\Identity

Public key

E:\temp\Identity.pub

After exiting the Wizard, upload the public key file to the appropriate folder on your SSH server. See help or refer to your SSH server documentation for more

Operations on the SSH server

When the key pair is generated on the SSH client, the SSH server (network device) copies the public key files into flash, and associated these public key files with SSH clients' username. Each username can be associated with one RSA public key and one DSA public key. As shown below:

```
Ruijie# configure terminal
Ruijie(config)# ip ssh peer test public-key rsa flash:rsa.pub
Ruijie(config)# ip ssh peer test public-key dsa flash:dsa.pub
```

By doing so, SSH clients can log into network devices through public key based authentication.

Transferring files through SSH

Operation on the SSH server:

SSH transfers files by means of the SCP protocol (Secure Copy). The client can upload files to the network device or download files from the network device through SCP. To realize such function, enable the SCP server function, as shown below.

```
Ruijie# configure terminal
Ruijie(config)# ip scp server enable
```

By doing so, client can connect server and transfer files through SCP. SCP server use SSH thread. So when a client connects the network device transferring files through SCP, it will takes up a VTY line. (when using the command show user, you can see the user type is SSH).

Operation on the SSH server:

Both Unix and Linux platform carries SCP command. Taking Ubuntu Linux as an example to demonstrate the use of SCP command, as shown below:

```
Grammar of SCP command:
    scp [-1246BCpqrv] [-c cipher] [-F ssh_config] [-i identity_file]
       [-l limit] [-o ssh_option] [-P port] [-S program]
       [[user@]host1:]file1 [...] [[user@]host2:]file2
Explanation for some options:
-1 : Use SSH version 1 (by default, use SSH version 2);
-2 :  Use SSH version 2 (by default);
-C :  Specify compressed transmission;
-c :  Specify encryption algorithm;
-r :  Transmit all files under this content;
-i :  Specify key pair;
-l :  Limit the transmission rate (measured by Kbits);
For other detailed parameters, see als the scp.0.
```

Take Ubuntu 7.10 as an example to demonstrate file transferring.

The designated user named *test copies config.text* file from a network device with an IP address 192.168.195.188 to the local /*root*. As shown below:

```
root@dhcpd:~# scp test@192.168.195.188:/config.text /root/config.text
test@192.168.195.188's password:
config.text                100% 1506    1.5KB/s   00:00
Read from remote host 192.168.195.188: Connection reset by peer
```

Note

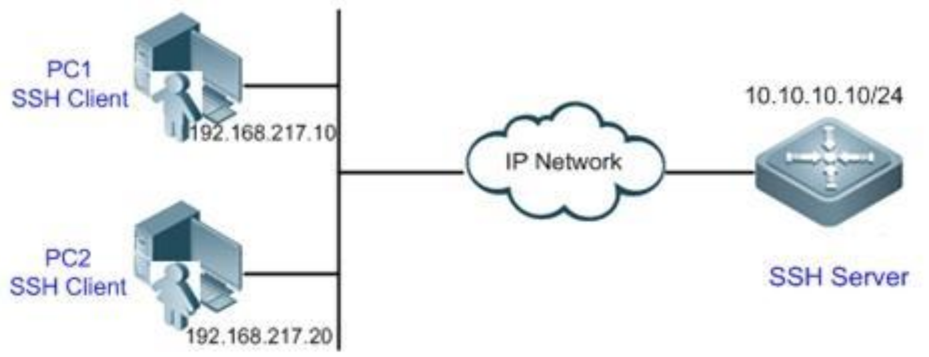
Most options are client related only. And few of these options require support from both client and server. The SCP server on Ruijie Networks' devices do not support -d -p –q –r options. So when user configuring these options, the device indicates unsupported message.

If no rate limitation (-l option) is configured in advance, the CPU usage will rise when the client downloading files from server, and recover after the downloading. The console remains available but other application tasks will be affected.

Typical SSH Configuration Examples

Configuring SSH Local Authentication

Figure 42 Networking diagram for SSH local password protection



Application Requirements

As shown in Figure , to ensure the security of information exchange, PC1 and PC2 serve as SSH clients from which users will log in to the SSH Server through SSH. The specific requirements are shown below:

- SSH users adopt line password authentication.
- Lines 0 to 4 are activated at the same time. The login password for line 0 is "passzero", and the login password for other four lines is "pass". Any user name can be used.

Notes

- Notes on SSH Server configuration are as follows:

98) Globally enable SSH Server. By default, SSH Server supports SSHv1 and SSHv2.
99) Configure a key. The SSH server will use this key to decrypt the encrypted passwords received from the SSH clients, and compare the plain text with the password stored on the server before returning a message that indicates a successful or failed authentication. SSHv1 uses RSA key, while SSHv2 uses RSA or DSA key.
100) Configure the IP address of the Gi 1/1 interface of the SSH server. SSH clients connect to the SSH server through this interface. The routes from SSH clients to the SSH server are reachable.

- Configurations on SSH Clients:

There are many SSH client software, such as Putty, Linux, OpenSSH and etc. In this example, SecureCRT is used as the SSH client software. For configuration details, see the "Configuration Steps" section.

Configuration Steps

- Configure the SSH Server

Before configuring relevant SSH features, make sure the routes from SSH clients to the SSH server are reachable. The IP addresses of respective interfaces are shown in Figure 6, and the IP address and route configuration are omitted herein.

Step 1: Enable SSH Server

```
Ruijie(config)# enable service ssh-server
```

Step 2: Generate an RSA key

```
Ruijie(config)#crypto key generate rsa
% You already have RSA keys.
% Do you really want to replace them? [yes/no]:
```

```
Choose the size of the key modulus in the range of 360 to 2048 for your
Signature Keys. Choosing a key modulus greater than 512 may take
a few minutes.

How many bits in the modulus [512]:
% Generating 512 bit RSA1 keys ...[ok]
% Generating 512 bit RSA keys ...[ok]
```

Step 3: Configure the IP address of Gi 1/1 interface. The client will use this IP address to connect to the SSH server.

```
Ruijie(config)#interface gigabitEthernet 1/1
Ruijie(config-if- gigabitEthernet 1/1)#ip address 10.10.10.10 255.255.255.0
Ruijie(config-if- gigabitEthernet 1/1)#exit
```

Step 4: Configure login passwords for lines

! Configure the login password for line 0 as "passzero"

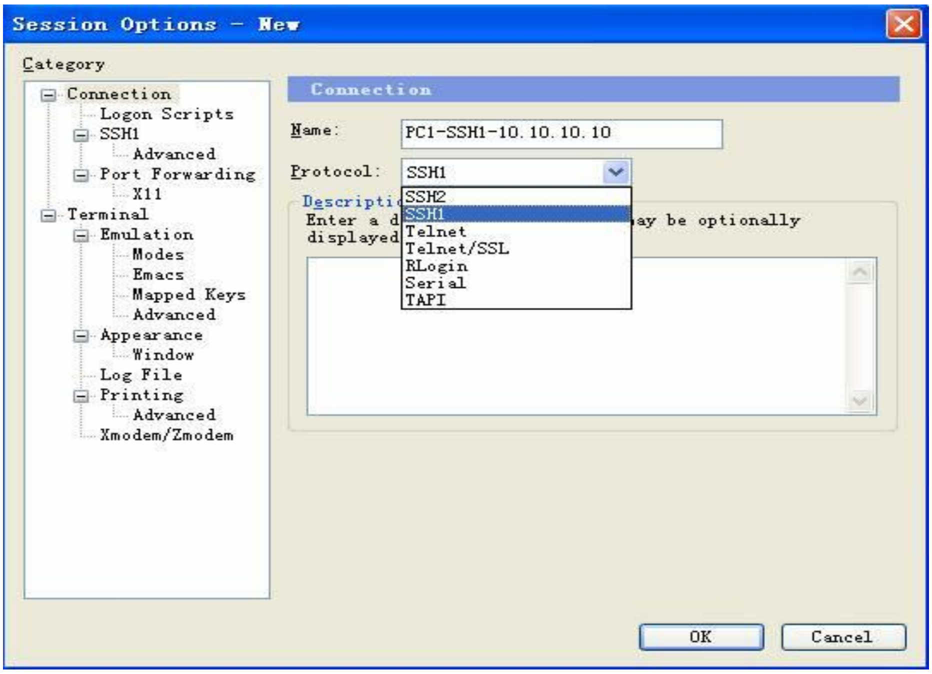
```
Ruijie(config)#line vty 0
Ruijie(config-line)#password passzero
Ruijie(config-line)#privilege level 15
Ruijie(config-line)#exit
```

! Configure the login password for lines 1 to 4 as "pass"

```
Ruijie(config)#line vty 1 4
Ruijie(config-line)#password pass
Ruijie(config-line)#privilege level 15
Ruijie(config-line)#exit
```

■ Configure SSH Clients (PC1 and PC2)

Start SecureCRT, as shown in the following figure. Use SSH1 for login authentication. Any session name can be specified (here the session name is configured as PC1-SSH1-10.10.10.10).

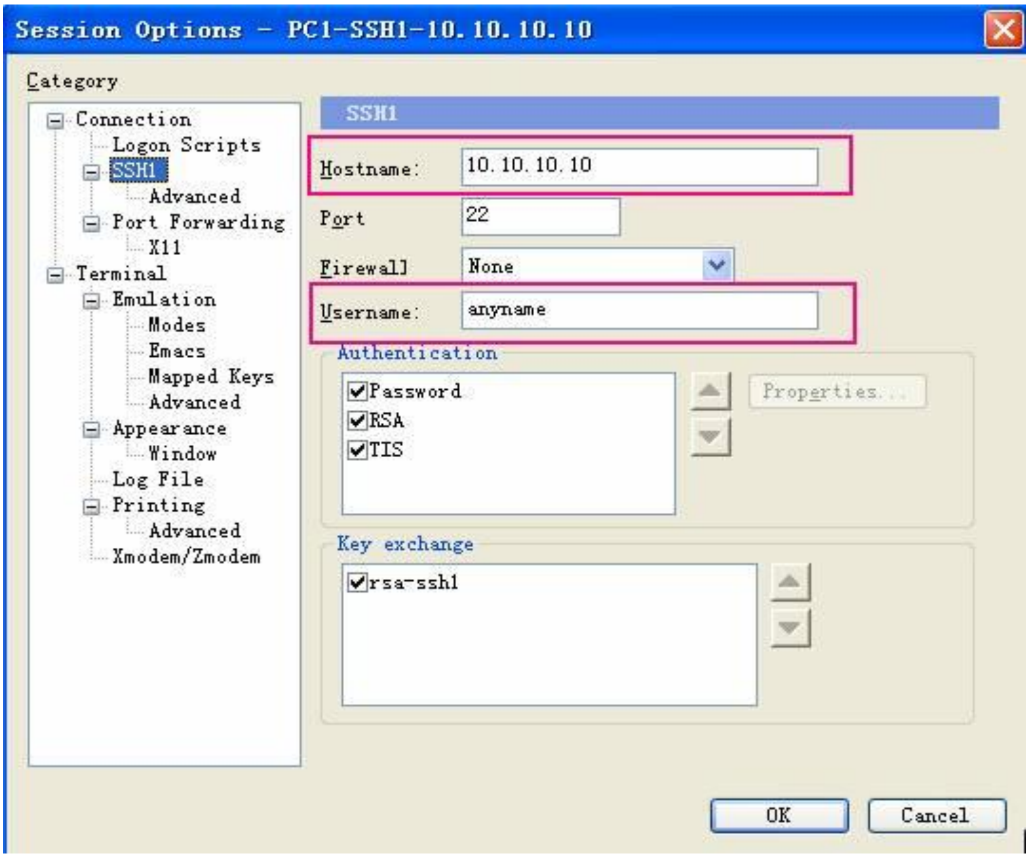


Name: PC1-SSH1-10.10.10.10

Protocol: SSH1

Description: Enter a d... ...ay be optionally displayed

Configure SSH attributes. The host name is the IP address of the SSH server (10.10.10.10 in this example). Since user name is not required, you can type in any user name in the **User Name** field, but this field cannot be left blank (the user name is **anyname** in this example).



Session Options - PC1-SSH1-10.10.10.10

Category

SSH1

Hostname: 10.10.10.10

Port 22

Firewall None

Username: anyname

Authentication

☑Password
☑RSA
☑TIS

Key exchange

☑rsa-ssh1

Verifying the Configuration

- Verify the SSH Server configuration

Step 1: Run the **show running-config** command to verify the current configuration:

```
Ruijie#show running-config

Building configuration...
!
enable secret 5 $1$eyy2$xs28FDw4s2q0tx97
enable service ssh-server
!
interface gigabitEthernet 1/1
 ip address 10.10.10.10 255.255.255.0
line vty 0
privilege level 15
 login
 password passzero
line vty 1 4
privilege level 15
```

```
 login
 password pass
!
end
```

■ Verify the configuration of SSH clients

Step 1: Establish a remote connection.

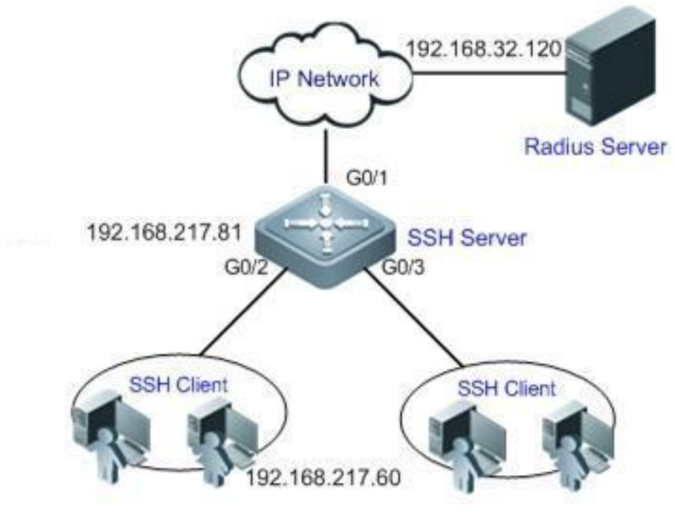
Establish a connection and type in the correct password in order to enter the interface of the SSH Server. The login password for line 0 is "passzero", and the login password for other four lines is "pass".

Step 2: Query the online user.

```
Ruijie#show users
    Line        User        Host(s)             Idle       Location
   0 con 0                  idle               00:03:16
   1 vty 0                  idle               00:02:16   192.168.217.10
*  2 vty 1                  idle               00:00:00   192.168.217.20
```

Example of Configuring SSH AAA Authentication

Figure 43 Networking diagram for SSH AAA authentication



Application Requirements

As shown in Figure 10, to ensure the security of information exchange, PCs serve as SSH clients from which users will log in to the SSH Server through SSH.

To better implement security management, SSH clients adopt AAA authentication. Meanwhile, for stability consideration, two authentication methods are configured in the AAA authentication method list: Radius server authentication and local authentication. Radius server authentication has a higher priority than local authentication unless no response is received during Radius server authentication.

Notes

- The routes from SSH clients to the SSH server and the route from the SSH server to the Radius server shall be reachable.
- SSH Server related configuration is complete on the network device. The configuration tips have been described in the previous example, and are not further described herein.
- AAA authentication related configuration is complete on the network device. AAA defines identity authentication and type by creating a method list, which is then applied to the specific service or interface. For details, see the "Configuration Steps" section.

Configuration Steps

The routes from SSH clients to the SSH server and the route from the SSH server to the Radius server shall be reachable. Route configuration will not be further described. For details, see the section about route configuration in this manual.

- Configure relevant SSH features on the network device

Step 1: Enable SSH Server

```
Ruijie(config)# enable service ssh-server
```

Step 2: Generate a key

! Generate an RSA key

```
Ruijie(config)#crypto key generate rsa
% You already have RSA keys.
% Do you really want to replace them? [yes/no]:
Choose the size of the key modulus in the range of 360 to 2048 for your
Signature Keys. Choosing a key modulus greater than 512 may take
a few minutes.
How many bits in the modulus [512]:
% Generating 512 bit RSA1 keys ...[ok]
% Generating 512 bit RSA keys ...[ok]
```

! Generate a DSA key

```
Ruijie(config)#crypto key generate dsa
Choose the size of the key modulus in the range of 360 to 2048 for your
Signature Keys. Choosing a key modulus greater than 512 may take
a few minutes.
How many bits in the modulus [512]:
% Generating 512 bit DSA keys ...[ok]
```

Step 3: Configure the IP address of the device. The clients will use this address to connect to the SSH server.

```
Ruijie(config)#interface gigabitEthernet 1/1
Ruijie(config-if-gigabitEthernet 1/1)#ip address 192.168.217.81 255.255.255.0
Ruijie(config-if-gigabitEthernet 1/1)#exit
```

- Configure AAA authentication on the network device

Step 1: Enable AAA on the device

```
Ruijie#configure terminal
Ruijie(config)#aaa new-model
```

Step 2: Configure information about the Radius server (the shared key used by the SSH server for communicating with the Radius server is "aaaradius"

```
Ruijie(config)#radius-server host 192.168.32.120
Ruijie(config)#radius-server key aaaradius
```

Step 3: Configure an AAA authentication method list

! Configure a login authentication method list (Radius server authentication followed by local authentication), and the name of the method list is "method".

```
Ruijie(config)#aaa authentication login method group radius local
```

Step 4: Apply this method list to the lines

```
Ruijie(config)#line vty 0 4
Ruijie(config-line)#login authentication method
Ruijie(config-line)#exit
```

Step 5: Configure a local user database

! Configure a local user database (configure the user name and password, and bind the user to a privilege level)

```
Ruijie(config)#username user1 privilege 1 password 111
Ruijie(config)#username user2 privilege 10 password 222
Ruijie(config)#username user3 privilege 15 password 333
```

! Configure a password for local Enable authentication

```
Ruijie(config)#enable secret w
```

Verifying the Configuration

Step 1: Run the **show running-config** command to verify the current configuration:

```
Ruijie#show run

aaa new-model
!
aaa authentication login method group radius local
!
username user1 password 111
username user2 password 222
username user2 privilege 10
username user3 password 333
username user3 privilege 15

no service password-encryption
!
```

```
radius-server host 192.168.32.120
radius-server key aaaradius
enable secret 5 $1$hbgz$ArCsyqty6yyzzp03
enable service ssh-server
!
interface gigabitEthernet 1/1
 no ip proxy-arp
 ip address 192.168.217.81 255.255.255.0
!
ip route 0.0.0.0 0.0.0.0 192.168.217.1
!
line con 0
line vty 0 4
 login authentication method
!
end
```

Step 2: Configure the Radius Server. This example describes how to configure the SAM server.

Choose **System Management** > **Device Management**, and type in 192.168.217.81 as the IP address of the device and the device key **aaaradius**;

Choose **Security Management** > **Device Management Privilege**, and configure a privilege level for the login user;

Choose **Security Management** > **Device Administrator**, and type in **user** as the user name and **pass** as the password.

Step 3: Establish a remote SSH connection on the PC.

For details about how to set SSH client software and establish a connection, see the previous example.

Type in **user** as the SSH user and **pass** as the password. The user will log in successfully.

Step 4: Query the online user.

```
Ruijie#show users
    Line        User        Host(s)             Idle      Location
   0 con 0                  idle               00:00:31
 * 1 vty 0     user         idle               00:00:33   192.168.217.60
```

Configuring IP Accounting

Understanding IP Accounting

Overview

IP accounting, an easy-to-use traffic management tool, classifies and collects statistics on the traffic passing routers by source IP address and destination IP address. The collected statistics include the number of packets and number of bytes. Traffic accounting and traffic analysis can be implemented on the basis of IP accounting statistics.

Configuring IP Accounting

Enabling IP Accounting

IP accounting is enabled on the inbound or outbound interface. You need to specify an interface and the direction when enabling IP accounting. In addition, you can configure a traffic classification rule while enabling IP accounting. The system will collect statistics on traffic based on the configured rule.

Use the following commands to enable IP accounting.

Command	Function
Ruijie> **enable**	Enters privileged EXEC mode.
Ruijie# **config terminall**	Enters global configuration mode.
Ruijie(config)# **interface** *interface-type interface-number*	Enters interface configuration mode.
Ruijie(config-if)# **ip accounting** { **ingress** | **egress**} **list** { *acl_list_num* | *acl_list_name* }	Enables IP accounting on the interface.
Ruijie(config-if)# **end**	Returns to privileged EXEC mode.
Ruijie# **copy running-config startup-config**	Saves the configuration.

To disable IP accounting on the interface, use the **no ip accounting** { **ingress** | **egress** } command in interface configuration mode.

Configuration example

```
Ruijie# config terminal
Ruijie(config)# interface gigabitEthernet 0/1
Ruijie(config-if)# ip accounting ingress list 20 //Enable IP Accounting on the 0/1 interface
to classify and collect statistics on incoming traffic based on ACL 20.
Ruijie(config-if)# exit
Ruijie(config)# interface gigabitEthernet 0/2
Ruijie(config-if)# ip accounting egress list 10 //Enable IP Accounting on the 0/2 interface
to classify and collect statistics on outgoing traffic based on ACL 10.
Ruijie(config-if)# exit
Ruijie(config)#
```

Displaying IP Accounting Configuration

Use the **show ip accounting config** command to query IP accounting configuration on an interface. For exampe:

```
Ruijie# show ip accounting config
GigabitEthernet 0/1
ip accounting ingress list 20
GigabitEthernet 0/1
ip accounting egress list 10
```

Displaying IP Accounting Statistics

Use the **show ip accounting interface** *interface-type interface-number* { **ingress** | **egress** } { **interior** | **exterior** } command to query IP accounting statistics on the inbound or outbound interface in priveledged, global, or interface mode, with **interior** indicating the statistics matching an ACL rule and **exterior** indicating the statistics not matching the ACL rule.

Command	Function
Ruijie> **enable**	Enters privileged EXEC mode.
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **interface** *interface-type interface-number*	Enters interface configuration mode.
Ruijie(config-if)# **show ip accounting** *interface interface-type interface-number* { **ingress** | **egress** } { **interior** | **exterior** }	Displays IP accounting statistics on the interface.
Ruijie(config-if)# **end**	Returns to privileged EXEC mode.

Configuration example

```
Ruijie# config terminal
Ruijie(config)# interface gigabitEthernet 0/1
Ruijie(config-if)# show ip accounting interface gigabitEthernet 0/1 ingress interior
```

Clearing IP Accounting Statistics

Use the **clear ip accounting interface** *interface-type interface-number* { **ingress** | **egress** } command to clear the IP accounting statistics on the specified inteface in priveledge EXEC mode.

The following example clears IP accounting statistics on the outbound interface.

```
Ruijie# clear ip accounting interface gigabitEthernet 0/1 egress
```

Configuring SDG

Understanding SDG

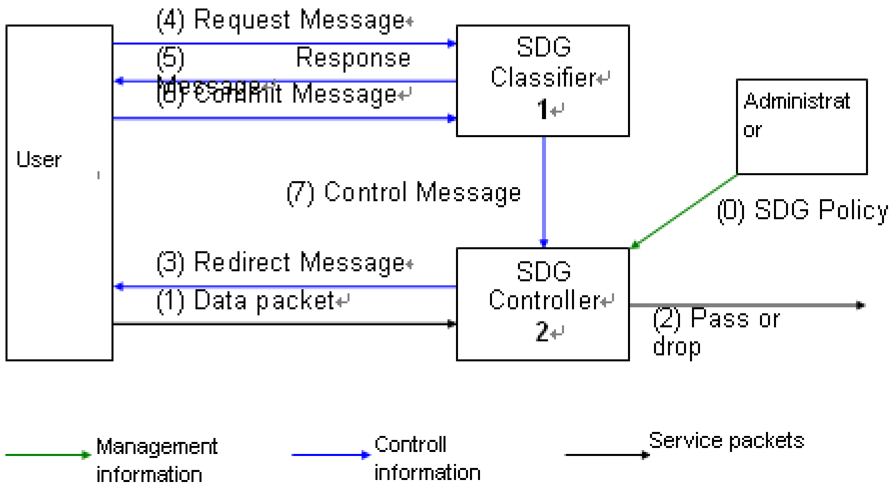
Overview of SDG

To be simple, Security Domain Gateway (SDG) achieves the logical isolation between different security domains. By limiting the user to access only the specified security domain at a time, users can be prevented from accessing different security domains and hence viruses will not spread to other protected domains, or important information will not leak to the insecure domains. SDG has two operating modes: local mode and linked mode.

Working Principle

Working principle of SDG in local mode:

Figure 44



As shown above, the SDG system consists of two parts:

Classifier: Through the interaction with users, the classifier determines the current role of each user, and sends this message to the controller. The classifier is realized by means of Web, so that the user does not need to install the client software.

Controller: The controller is mainly responsible for receiving the user role message sent by the classifier and applying access control to the traffic sent by the user as per the access permission assigned to such role. In addition, the controller is also responsible for triggering a message prompting the user to reselect a role while accessing an unauthorized domain.

User: During the interaction with SDG, the user sends traffic to the controller on one hand and negotiates with the Classifier for role selection on the other hand.

Administrator: It is responsible for establishing SDG policies on the controller. Such a policy determines the access permissions of different user roles, implementing "logical isolation". The policy is generally configured through CLI.

The working procedures of SDG are shown below:

(0): The administrator establishes a SDG policy according to actual needs, covering the user role, isolated domain and the access control rules applied between user roles and isolated domains.

(1) → (2): The Controller receives the traffic sent by the user and applies the access control, so that valid packets can pass and invalid packets failing to trigger "user role selection" will be discarded. A criterion for identifying valid packet depends on an SDG policy: the user role sending this packet is allowed to access the isolated domain.

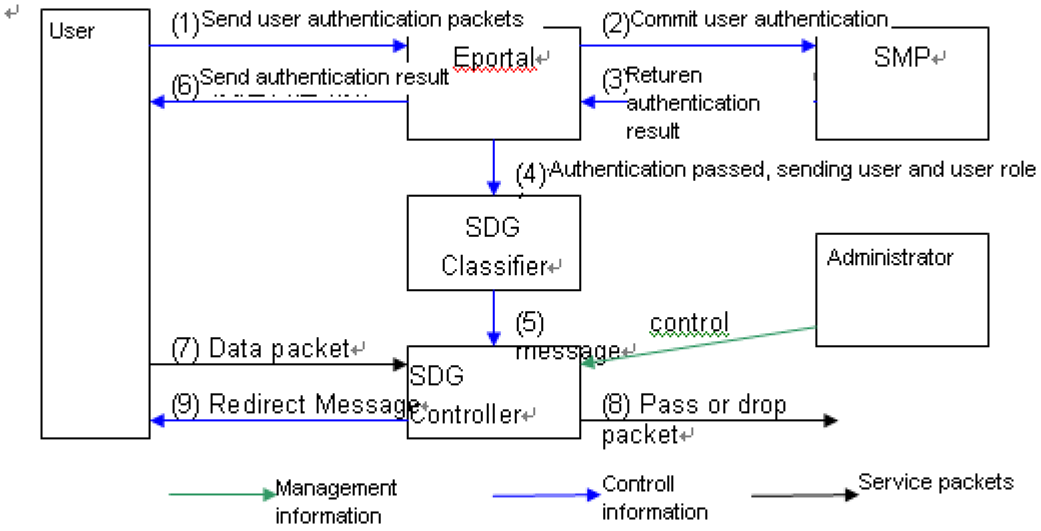
(1) → (3) → (4): As for the invalid packets sent by the user, if such packets are capable of triggering "role selection", then the controller will send an Http redirection packet to the user and redirect the user request to the "user role selection" page on the Classifier. One condition for triggering "user role selection" is that the packet must be a valid http request packet.

(4) → (5) → (6): The user proactively (or through Controller redirection) sends the "user role selection" request to the Classifier, which will reply to the user (browser) with this page. No matter which role is selected by the user on the page, the selection result will be submitted to the Classifier.

(7): The Classifier forwards the user role selection result to the Controller, which will record and use such information as the reference for security check of this user during future access.

Working principle of SDG in linked mode

Figure 45



Eportal: Eportal server

SMP: security authentication server

Other flows are the same as local mode.

The working procedures of SDG are shown below:

(0): The administrator establishes an SDG policy according to actual needs, covering the user role, isolated domain and access control rules applied between user roles and isolated domains.

(1)→(2): The user sends authentication packet to Eportal, which will forward the user authentication packet to SMP.

(3) MP verifies user information and sends the authentication result to Eportal.

(4)→(5): In case of successful authentication, Eportal will send the user and user role to the SDG Classifier, which will update the user role and forward the user role selection result to the Controller. The Controller will record and use such information as the reference for security check of this user during future access.

(6): Eportal will send the authentication result to the user, which will reselect the role or send traffic according to the authentication result.

(7)→(8)→(9): The SDG Controller matches the user role. If the user role does not match and the redirection condition is met, it will send a redirection packet. The user will reselect the role and send a user authentication packet to the Eportal, or else it will discard the packet. If matched, SDG check is passed.

Protocol Specification

N/A

Default Configurations

N/A

Configuring SDG

Configuring SDG Mode

While deploying SDG, you can select different SDG operating modes according to users' security needs. The SDG working in local mode is easy to deploy and will be ready to operate after the router is configured. To deploy SDG working in linked mode, you need to add an SMP server. This mode, however, supports user authentication and hence provides higher security.

To configure SDG mode, run the following commands:

Command	Function
Ruijie> **enable**	Enters privileged command mode.
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **ip sdg mode** *link*	Enables SDG linked mode.

Configuring the IP Address of the SMP Server

In linked mode, you need to configure the IP address of the SMP server for redirection. The user can also use an SMP URL for redirection.

To configure the IP address of the SMP server, run the following commands:

Command	Function
Ruijie> **enable**	Enters privileged command mode.
Ruijie# **config terminal**	Enters global configuration mode.

Ruijie(config)# **ip sdg portal** *ip* [url]	Configures the IP address for connecting to the Eport, namely the IP address of the Eportal server .

Configuring Aging Time for Offline Users

In linked mode, you need to configure aging time for users. A user is considered offline if the user does not initiate a connection request within the aging time.

To configure aging time for offline users, run the following commands:

Command	Function
Ruijie> **enable**	Enters privileged command mode.
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **ip sdg user-timeout** *time*	Configures aging time for offline users.

Configuring Keepalive Duration and Threshold of User Traffic

In association mode, the keepalive duration and threshold of user traffic must be configured. After the keepalive duration and threshold of user traffic are configured, the SDG controller checks whether the user traffic is lower than the configured traffic threshold within the configured keepalive duration. If the user traffic is lower than the configured traffic threshold within the keepalive duration, the user is considered offline. In this case, a trap message is sent to the SMP server to force the user to go offline.

Run the following commands to configure the keepalive duration and threshold of user traffic:

Command	Function
Ruijie> **enable**	Enters the privileged EXEC mode.
Ruijie# **config terminal**	Enters the global configuration mode.
Ruijie(config)# **ip sdg offline-detect idle-timeout** *time-out* **threshold** *flow-num*	Configures the keepalive duration and threshold of user traffic.

Configuring DNS Hijacking

In association mode, DNS hijacking must be configured. DNS hijacking is not configured by default and will be immediately enabled once it is configured. When **action** is **drop**, all DNS query packets except Class A DNS query packets are discarded. When **action** is **permit**, all DNS query except class A query packets are forwarded.

Run the following commands to configure DNS hijacking:

Command	Function
Ruijie> **enable**	Enters the privileged EXEC mode.
Ruijie# **config terminal**	Enters the global configuration mode.
Ruijie(config)# **ip sdg dns-hijack interface loopback** *num* **action** [**drop** | **permit**]	Configures DNS hijacking.

Configure the Default User

In local mode, add the default user into the specified user group. Upon the first access, the user has the permission to access the specified user group.

To configure the default user, run the following commands:

Command	Function
Ruijie> **enable**	Enters privileged command mode.
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)#**ip sdg permit-user** *user-ip user-mask* **user-group** *group-name*	Adds the default user.
Ruijie(config)#**no ip sdg permit-user** *user-ip user-mask* **user-group** *group-name*	Removes the default user.

Configuring SDG Classifiers

To control SDG, you must define SDG classifiers. Each SDG classifier defines a series of user groups (user roles). One user belongs to only one user group at the same time.

The SDG classifier created is applied to SDG policies. When user access violates the SDG policy, the user selection page will be displayed to prompt the user to select a user group.

The user can also take the initiative to access the user selection page to select a user group. The URL of the user selection page is:

"http://" + *device interface address* + "/sdg" + *classifier ID* + ".htm?ruijie_query_id=sdg". For example, if the interface address is 192.168.52.52 and the classifier ID is 1, then the corresponding URL is:

http://192.168.52.52/sdg001.htm

To configure SDG classifiers, run the following commands:

Command	Function
Ruijie> **enable**	Enters privileged command mode.
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **ip sdg classifier** *classifier-id*	Configures SDG classifiers and enters SDG classifier command mode.

Caution Web server must be enabled in order to generate the user role selection page.

```
Ruijie(config)# enable service web-server
```

Configuring User Groups

After creating SDG classifiers, run the **user-group** command to configure user groups to be included in this classifier.

To configure user groups, run the following command:

Command	Function
Ruijie> **enable**	Enters privileged command mode.
Ruijie# **config terminal**	Enters global configuration mode.

Ruijie(config)# **ip sdg classifier** *classifier-id*	Configures SDG classifiers and enters SDG classifier command mode.
Ruijie(config)# **user-group** *group-name*	Configures the user group included in an SDG classifier.

Configuring Static Users

In local mode, add users into the specified user group. The statically configured user will not be changed while the user is selecting a role.

Command	**Function**
Ruijie> **enable**	Enters privileged command mode.
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **user-group** *group-name*	Configures user groups to be included in an SDG classifier.
Ruijie (config-user-group)# **user** ip	Adds a static user.
Ruijie (config-user-group)#**no user** ip	Removes a static user.

Clearing Users in a User Group

In local mode, remove non-static users from a user group; in linked mode, remove all users from a user group.

Command	**Function**
Ruijie> **enable**	Enters privileged command mode.
Ruijie# **clear user-group** *group-name*	Removes users from a user group.

Configuring SDG Control Policies

A SDG policy can be configured in either inbound or outbound direction of an interface. It consists of one ACL and one SDG classifier. The ACL is used to define the isolation policy, which must be based on the user groups included in the SDG classifier. When user access violates the isolation policy, the user selection page defined in SDG classifier will be displayed to prompt the user to select an appropriate user group.

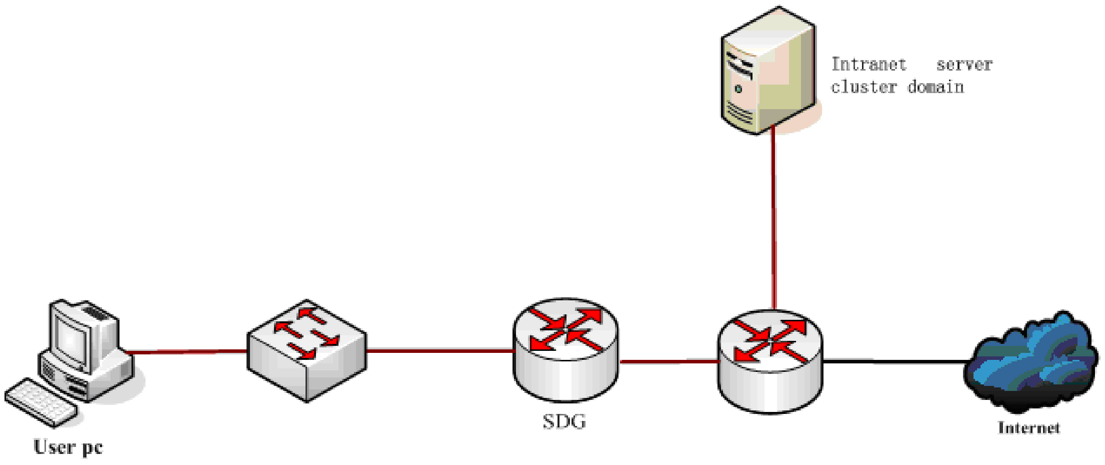
To configure SDG control policies, run the following commands:

Command	**Function**
Ruijie> **enable**	Enters privileged command mode.
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)#**interface** *gigabitEthernet 0/0*	Enters interface configuration mode.
Ruijie(config-if-gigabitEthernet 0/0)#**ip sdg in|out access-group** *acl-no* **trigger** *classifier-id*	Configures an SDG control policy.

Configuration Examples

Local mode

Figure 46



As shown above, after transparent bridge mode is enabled on the SDG device, SDG functions can be realized without changing the network structure.

Configuration:

101) Configure SDG to work in local mode

Command	Function
Ruijie(config)# **ip sdg mode local**	Configures SDG to work in local mode.

102) Configure the SMP address for SDG in linked mode

Command	Function
Ruijie(config)# **ip sdg portal** 10.1.1.2 http:// *xxx.xxx.xxx/eportal*	Configure the IP address for connecting to the Eport, namely the IP address of the Eportal server.

103) Create a user group

Define a user group for each user role, including intranet user group (intranet_user) and internet user group (internet_user).

Command	Function
Ruijie(config)# **user-group** *intranet_user*	Configures the intranet user group.
Ruijie(config)# **user-group** *internet_user*	Configures the internet user group.

104) Configure a static user

In local mode, add a user into the user group

Command	Function
Ruijie(config-user-group)#**user** 192.168.50.18	Adds a user.
Ruijie(config-user-group)# **no user** 192.168.50.18	Removes a user.

105) Create a domain

An intranet server cluster domain (intranet_site) needs to be defined.

Command	Function

Ruijie(config)# **network-region** *intranet _site*	Configures an intranet server cluster domain.
Ruijie(config-network-region)# **network** *192.168.0.0 255.255.0.0* …………	Configures all network segments included in the intranet server cluster domain.

106) Configure an SDG classifier

Define an SDG classifier, which shall include intranet user group (intranet_user) and internet user group (internet_user).

Command	**Function**
Ruijie(config)# **ip sdg classifier** *1*	Defines SDG classifier 1.
Ruijie(config-sdg-classifier)# **user-group** *intranet_user*	Adds intranet_user into the classifier.
Ruijie(config-sdg-classifier)# **user-group** *internet_user*	Adds internet_user into the classifier.

107) Configure an isolated access policy

Use an ACL to define an isolation policy.

Command	**Function**
Ruijie(config)# **ip access-list extend** *100*	Configures extended ACL 100.
Ruijie(config-ext-acl)# **permit ip user-group** *intranet_user* **network-region** *intranet_site*	Allows intranet users to access intranet servers.
Ruijie(config-ext-acl)# **deny ip user-group** *intranet_user* **any**	Prohibits intranet users from accessing other sites.
Ruijie(config-ext-acl)# **deny ip user-group** *internet_user* **network-region** *intranet_site*	Prohibits Internet users from accessing intranet servers.
Ruijie(config-ext-acl)# **permit ip user-group** *internet_user* **any**	Allows Internet users to access other sites.

Note: DNS traffic must not be confined. You can create ACEs at the beginning of the ACL to permit DNS traffic.

108) Configure an SDG policy

Configure an SDG policy on the interface. The SDG policy is associated with one ACL and one SDG trigger. HTTP requests violating this ACL will be redirected to the page configured by the SDG trigger.

Command	**Function**
Ruijie(config)# **Interface** *gi 0/0*	# Enters Gi 0/0 interface.
Ruijie(config-interface)# **ip sdg in access-group** *100* **trigge**r *1*	Configures an SDG policy in the inbound direction and associates the policy with ACL 100 and SDG classifier 1.

109) Enable web-server

Command	**Function**
Ruijie(config)# **enable service web-server**	The SDG user selection page will only be generated only after web-server is enabled.

110) Enable transparent bridge mode

Command	**Function**

Ruijie(config)# transparent	Enables transparent bridge mode.

Note: Only fast forwarding supports transparent bridge mode

111) Authenticating a user

You can directly type the URL of the SMP server in the address box of a Internet Explorer for authentication, or makes SDG trigger authentication upon denial of access.

Configuring Anti-attack Features on Devices

Overview of Device Anti-attack

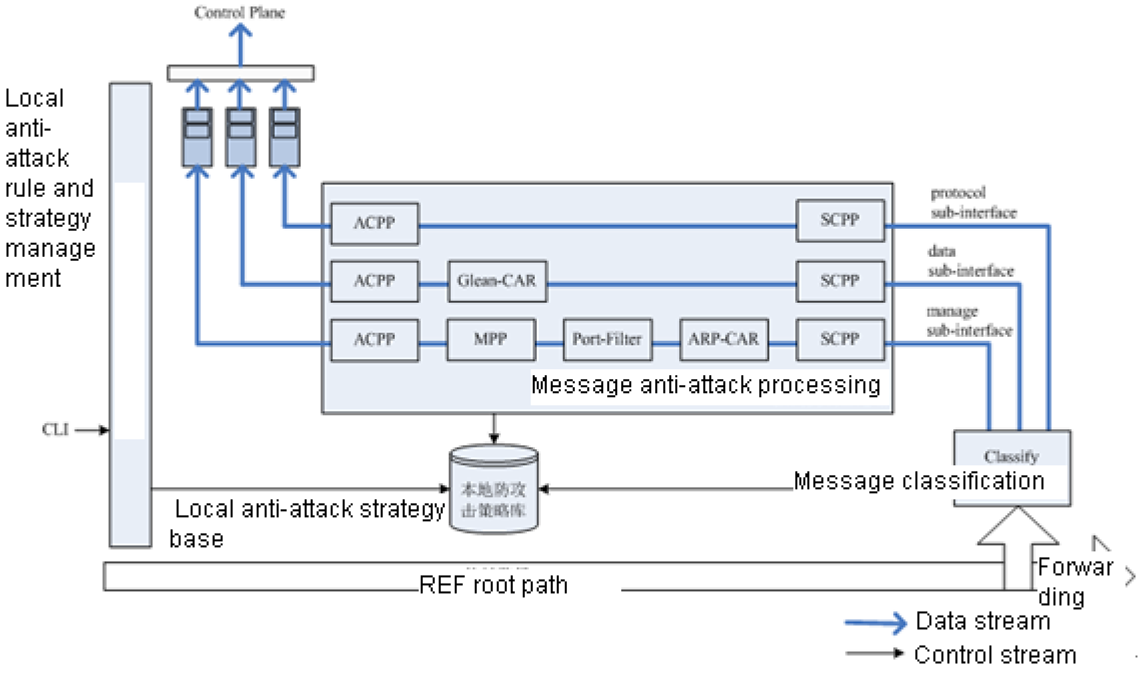
When encountering a network attack or heavy traffic, the device on a complex network may report the following exceptions:

112) Extremely high CPU usage;
113) Slow response or no response of CLI;
114) Loss of link or network control protocol messages, consequently leading to link or network delay variation;
115) Unauthorized occupation of bandwidth, resulting in the failure to process important protocol messages.

Such phenomena are due to the difference in processing capacity of control and forwarding planes on one hand and the lack of protection of the control plane on the other hand. The anti-attack module is to classify, filter and rate-limit the data messages that need to be forwarded to the control layer for processing, protecting the key resources of the control plane.

The following figure illustrates the principle and process of device anti-attack:

Figure 47



As shown in the figure, device anti-attack consists of numerous sub-modules:

Classify: Identify and classify the data traffic destined for the control plane. There are three categories of traffic including protocol, manage and data. The sub-module sets a base for subsequent rate limiting and filtering.

Sub-interface: The three traffic categories correspond to the following three sub-interfaces. The three sub-interfaces and streams through these interfaces are defined as follows:

Protocol sub-interface: All protocol control streams sent to the local device, such as link layer protocol messages, routing protocol messages, etc.

Manage sub-interface: All management protocol streams sent to the local device, such as FTP, TELNET, SNMP streams, etc. In addition, ARP and ICMP traffic also falls into this category.

Data sub-interface: All data streams that cannot be processed by any REF plane.

Note The sub-interface in this manual is different from what is usually regarded. It only represents an internal path through which a type of traffic is sent to the control plane, helping you configure anti-attack and process traffic.

SCPP: protects the control plane by subdividing traffic. SCPP delivers more delicate rate limiting and protection according to user-defined policies.

Glean-CAR: limits the rate of the traffic to the REF plane matching REF Glean adjacency (traffic with a direct route but without a host route matching the destination IP address is diverted to the control plane for destination IP address resolution).

ARP-CAR: Since the REF plane cannot complete processing ARP messages, these messages must be forwarded to the control plane. ARP-CAR can limit the rate of ARP messages from each neighbor.

Port-Filter: checks whether ports have been enabled for local TCP and UDP messages and filters network traffic for which no local network service is enabled.

MPP: Management Plane Protection (MPP) allows administrators to specify one or more interfaces as in-band management interfaces (receiving management messages and forwarding normal service messages). After the MPP function is enabled, only specified in-band management interfaces are allowed to receive the management messages of specified protocol. However, service messages, protocol messages, ARP messages, etc are not affected.

ACPP: Aggregate Control Plane Protection (ACPP) limits, on the basis of classification result by Classify, the rate of traffic on protocol sub-interface, manage sub-interface and data sub-interface with default or users' custom-made rate in order to ensure the traffic does not exceed the processing capacity of the control plane and the control planed is protected as a result.

Configuring Device Anti-attack

Device Anti-attack Configuration Tasks

Note Sub-functions, such as SCPP, Glean-CAR, ARP-CAR, Port-Filter, MPP, and ACPP, are independent from each other. Users may combine and configure them according to their needs and strategies. But it should be noticed that some sub-functions can be applied only to specified sub-interfaces.

Entering Control-plane Configuration Mode

All device anti-attack functions are configured in control-plane configuration mode. To enter control-plane configuration mode, run the following command:

Command	Function
Ruijie(config)# **control-plane** {**protocol** | **manage** | **data**}	Enters control-plane configuration mode and accesses corresponding sub-interfaces.

To exit control-plane configuration mode, run the **exit** command.

Configuring SCPP

SCPP can be used to distinguish and limit the rate of traffic in sub-interfaces according to user-defined policies. Rate limiting is classified into connection limiting and traffic bandwidth limiting. To configure SCPP, run the following commands:

Command	Function
Ruijie(config-cp)# **scpp list** *acl_no* {**bw-rate** bw-rate **bw-burst-rate** *bw-burst-rate* | **conn-total** *conn-num* | **conn-create-rate** conn-*create-rate* **conn-create-burst-rate** *conn-create-burst-rate*}	Configures a SCPP traffic bandwidth limit (unit: pps), a connection limit, etc for the traffic that complies with **acl_no** strategy on the sub-interface. *Acl_no*: ACL rule used to select the traffic in need of SCPP processing. *Bw-rate*: rate limit (unit: pps) *Bw- burst-rate*: burst rate limit (unit: pps) *Conn-num*: limit on the total number of connections *Conn-create-rate*: limit on the rate of connection establishment (unit: connection/s) *Conn-create-burst-rate*: limit on the burst rate of connection establishment (unit: connection/s)
Ruijie(config-cp)# **no scpp list** acl_no	Deletes configured SCPP rules.

SCPP processing can be applied to traffic on all sub-interfaces.

SCPP is disabled by default and will not be enabled until users configure SCPP rules explicitly.

Configuring Glean-CAR

For traffic that has a direct route but does not have its destination IP address resolved, configure Glean-CAR to limit the rate by using the following commands:

Command	Function
Ruijie(config-cp)# **glean-car** *packet_rate_per_group*	Configures a rate limit on Glean adjacency traffic initiated by users hashed to the same group *Packet_rate_per_group:* rate limit (unit: pps)
Ruijie(config-cp)# **no glean-car**	Deletes Glean-CAR rules.

The Glean-CAR function can only be configured on the data sub-interface.

Currently, the hash algorithm extracts the least significant *n* bits (*n* is determined by products) of source address.

It should be noticed that Glean-CAR can limit the rate of Glean adjacency matching traffic initiated by users hashed to the same group. For example, both user of A (192.168.52.57) and user B (192.168.60.57) (with the same hashed result) send traffic to destination host C (172.16.0.5) directly connected to the device. Before ARP messages of host C has been successfully resolved, only a maximum of five messages can be sent by users A and B to the control plane every second for destination IP ARP resolution if glean-car 5 is configured.

Glean-CAR is enabled by default and the rate limit of Glean adjacency matching traffic initiated by users (source) hashed to the same group is configured at 5 pps.

Configuring ARP-CAR

Run the following commands to configure ARP-CAR on ARP traffic reaching the local device.

Command	**Function**
Ruijie(config-cp)# **arp-car** *packet_rate_per_group*	Configures a rate limit of the ARP traffic initiated by users hashed to the same group. *Packet_rate_per_group:* rate limit (unit: pps)
Ruijie(config-cp)# **no arp-car**	Deletes ARP-CAR rules.

The ARP-CAR function can only be configured on the manage sub-interface.

Currently, the hash algorithm extracts the least significant *n* bits (*n* is determined by products) of source address.

It should be noticed that ARP-CAR can limit the rate of ARP traffic initiated by users hashed to the same group. For example, both user A (192.168.52.57) and user B (192.168.60.57) (with the same hashed result) initiate ARP requests of 192.168.52.1 to the device. If ARP-CAR 5 is configured, only a maximum of five messages can be sent by users A and B to the control plane every second for ARP response.

ARP-CAR is enabled by default and the rate limit of the ARP traffic initiated by users (source) hashed to the same group is configured at 5 pps.

Configuring Port-Filter

Run the following commands to configure Port-Filter to filter the transport layer messages that reach the local device yet have no services enabled:

Command	**Function**
Ruijie(config-cp)# **port-filter**	Enables Port-Filter.
Ruijie(config-cp)# **no port-filter**	Disables Port-Filter.

The Port-Filter function can only be configured on the manage sub-interface.

Port-Filter is disabled by default and will not be enabled until users configure Port-Filter rules explicitly..

Configuring MPP

Run the following commands to configure the in-band management interface and the management protocol messages the interface is allowed to receive:

Command	**Function**

Command	Function
Ruijie(config-cp)# **management-interface** *interface* **allow** {**ftp** | **http** | **https** | **ssh** | **snmp** | **telnet** | **tftp**}	Specifies the in-band management interface and configures the management protocol messages supported by the interface *Interface:* in-band management interface
Ruijie(config-cp)# **no management**-**interface** *interface*	Deletes the specified in-band management interface. The MPP function will be disabled if all in-band management interfaces are deleted.

The MPP sub-function can only be configured on the manage sub-interface.

A maximum of 16 in-band management interfaces can be configured and each of them can receive several or all management protocol messages.

After the MPP function is enabled, in-band management interfaces can receive specified management protocol messages and other interfaces do not receive management protocol messages.

MPP is disabled by default and will not be enabled until users configure MPP rules explicitly.

Configuring ACPP

Run the following commands to configure ACPP for the classisified traffic reaching the control plane:

Command	Function
Ruijie(config-cp)# **acpp bw-rate** *rate* **bw-burst-rate** *burst-rate*	Configures ACPP rules for the traffic on sub-interfaces. *Rate:* rate limit (unit: pps) *Burst-rate:* burst rate limit (unit: pps)
Ruijie(config-cp)# **no acpp**	Deletes ACPP rules.

ACPP can be applied to all sub-interfaces.

ACPP is disabled by default and will not be enabled until users configure ACPP rules explicitly.

Enabling Device Anti-attack with Default Rules

Run the following commands in control-plane configuration mode to configure device anti-attack with default rules:

Command	Function
Ruijie(config-cp)# **ef-rnfp enable**	Enables device anti-attack with default rules.
Ruijie(config-cp)# **ef-rnfp disable**	Disables device anti-attack.

Default device anti-attack rules and strategies are configured according to different products and platforms.

Maintaining Device Anti-attack

Maintenance of device anti-attack

To query configurations and statistics of device anti-attack, run the following command:

Command	Function

Ruijie# **show ef-rnfp** { **acpp** {**data** | **manage** | **protocol**}| **scpp** {**data** | **manage** | **protocol**}| **glean-car** | **arp-car** | **port-filter** | **mpp** | **all** }	Views configurations and statistics of device anti-attack.

You can use the **show ef-rnfp { acpp {data | manage | protocol}| scpp {data | manage | protocol}| glean-car | arp-car | port-filter | mpp | all }** command to query configurations and statistics of sub-functions, or the **show ef-rnfp all** command to query the configurations and statistics of available anti-attack functions.

Typical Device Anti-Attack Configuration Example

The following example shows the typical configuration of device anti-attack:

```
Ruijie# config
```

//Enter control-plane configuration mode and protocol sub-interface

```
Ruijie(config)# control-plane protocol
```

// Configure ACPP to set the traffic rate to 500 pps and burst traffic rate to 600 pps on the protocol sub-interface

```
Ruijie(config-cp)# acpp bw-rate 500 bw-burst-rate 600
```

//Enter control-plane configuration mode and data sub-interface

```
Ruijie(config)# control-plane data
```

// Configure ACPP to set the traffic rate to 500 pps and burst traffic rate to 600 pps on the data sub-interface

```
Ruijie(config-cp)# acpp bw-rate 500 bw-burst-rate 600
```

// Configure Glean-CAR, allowing 10 messages from a source to match Glean adjacency per second

```
Ruijie(config-cp)# glean-car 10
```

//Enter control-plane configuration mode and manage sub-interface

```
Ruijie(config)# control-plane manage
```

// Configuring ACPP to set the traffic rate to 500 pps and burst traffic rate to 600 pps on the manage sub-interface

```
Ruijie(config-cp)# acpp bw-rate 500 bw-burst-rate 600
```

// Configure ARP-CAR allowing 10 messages from a source to match Glean adjacency per second

```
Ruijie(config-cp)# arp-car 10
```

// Enable Port-Filter

```
Ruijie(config-cp)# port-filter
```

//Configure MPP rules by specifying gi0/0 interface as the in-band management interface and only allowing it to receive Telnet and SNMP messages

```
Ruijie(config-cp)# management-interface gi0/0 allow telnet snmp
```

Configuring RPL

Understanding RPL

Overview

Reverse path limited (RPL) enables packets to be sent and returned along the same path, ensuring that these packets are not discarded by firewalls that do not allow one-way connection.

When establishing a flow table, a router buffers the inbound port No. of the first packet. Reply packets of a data flow are matched to the flow table preferentially and are forwarded through the inbound port. (Both the routing table and policy-based routing table are not used preferentially, which is similar to the state-based forwarding mechanism of firewalls.)

Basic Concept

RPL

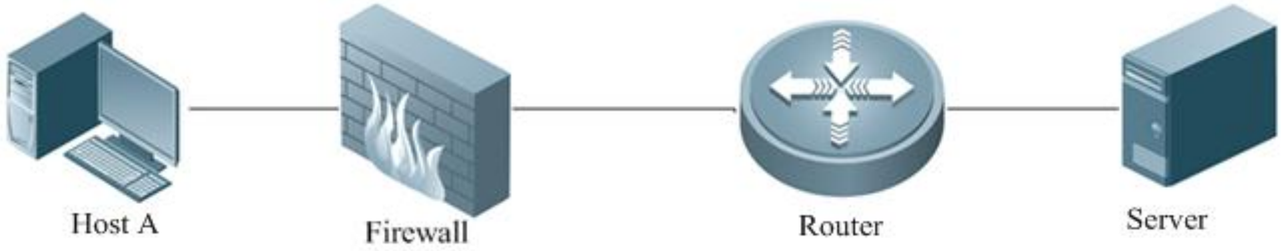
Work Principle

The source port for obtaining a packet is used as the outbound port for replying to the packet.

Typical Application

Figure 4 Networking topology for a terminal to access the server by traversing the firewall

Figure 48



Configuring RPL

Default Configuration

The following table describes the default configuration of RPL.

Feature	Default Setting
RPL	Disabled.

Configuring RPL

Command	Function
Ruijie(config-GigabitEthernet 0/0)#**ip reverse-path** [**access-list**] [*acl_id*]	Enables RPL. The value range of acl_id is as follows: 1 to 99 (IP standard access list) 100 to 199 (IP extended access list) 1300 to 1999 (IP standard access list, expanded range) 2000 to 2699 (IP extended access list, expanded range)
Ruijie(config-GigabitEthernet 0/0)#**no ip reverse-path**	Disables RPL.

Caution

Before enabling RPL on a subinterface, ensure that the subinterface and its peer subinterface are interconnected and the MAC address of the peer subinterface is learnt by this subinterface. If not, the one-way audio failure will occur. You can ping the IP address of the peer subinterface, or use the **shutdown** command to disable the primary interface of the subinterface and then use the **no shutdown** command to enable the primary interface. The preceding restriction does not apply when RPL is configured on other interfaces.

☑ This command is supported by routers only.

Configuration example`# Enable RPL.`

This example shows how to enable RPL on an interface.

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)#interface gigabitethernet 0/0/0
Ruijie(config-if)# ip reverse-path
Ruijie(config-if)#exit
Ruijie(config)#end
```

Displaying Device Configurations

Use the following command to show device configurations.

Command	Function
show running-config	Displays device configurations.

Configuration Examples

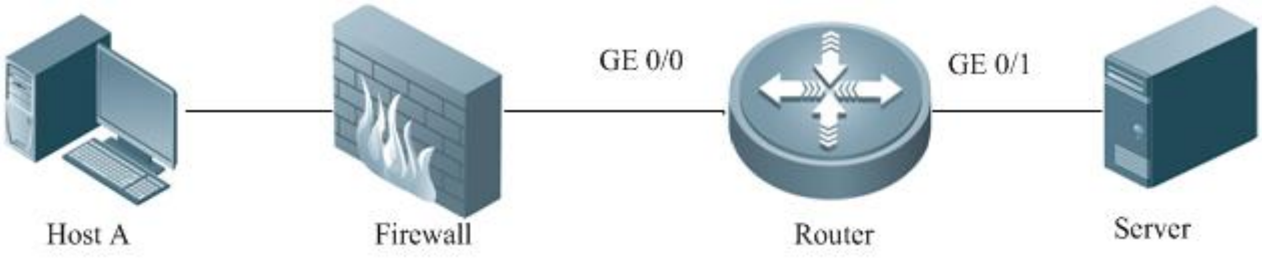
RPL Configuration Example

Networking requirements

Terminals of a network need to traverse the firewall to access the server.

Networking topology

Figure 49 Networking topology for a terminal to access the server through the router and firewall



Configuration Tips

- The router and host A is interconnected.
- IP forwarding is enabled on GE 0/0 and GE 0/1.
- Routes to host A are not required on the router.

Configuration Steps

Enter configuration commands in interface mode to configure RPL.

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)#interface gigabitethernet 0/0/0
Ruijie(config-if)# ip reverse-path
Ruijie(config-if)#exit
Ruijie(config)#end
```

Verification

Run the **show running-config** command on the router. **ip reverse-path** is displayed for GE 0/0. Ping the IP address of the server from host A. The server can be successfully pinged.

Configuring QoS

Configuring QOS

What is QOS

In a traditional IP network, the router treats all packets in the same way on a First In First Out (FIFO) basis and sends them to their destinations at best effort. However, it does not provide any assurance for the performances such as reliability and transmission delay of the packets.

As Internet becomes increasingly popular worldwide and IT is more widely used in social activities, people demand more and more from the network. IT needs evolve from the pure data information to the interactive multimedia information, from separate service to integrated transmission of data, voice and image services on a single network. More and more voice and image and other important data that require low bandwidth delay and be jitter sensitive and highly real time are transmitted over the network. On one hand, network resources become much diversified. On the other hand, the assurance of the service quality of the network incurs an important problem since the data, voice and image services have different requirements in delay, throughput or packet loss rate.

One solution to this problem is to increase the network bandwidth, which however is limited and costly as well. Other means to assure the service quality include the use of the techniques such as Policy-Based Routing, Congestion Management, Congestion Avoidance, Traffic Shaping and transmission compression to manage the traffic on the network and meet the needs of the increasing traffic on the network.

QoS (Quality of Service) is the ability of a network to provide better services for the specified network communications by using various basic technologies. To put it in simple way, different network service qualities are provided to suit the specific requirements: better qualities for the packets that are highly real-time and important; lower qualities for the common packets that do not need to be real time. To bear various services on the network, the network must be able to not only provide individual services but also offer different QoSs for them. It can be said for sure that QoS is a basic requirement for future IP networks.

Ruijie devices implement various QoS policies to meet the needs of different services for different QoSs.

Why QoS

QoS allows the network provides services for various network applications and communications in a proactive way. On the network, the QoS can be used to:

- Control resources: Users can control the network resources being used. For example, users can exercise control over the network resources occupied by FTP (File Transfer Protocol) transmission or assign higher priority for important data access.
- Sub-divide services. An ISP as a user can provide services of different QoS for different customers and packets of different requirements.
- In a network environment, different QoSs are provided for different applications to ensure network services for important packets. For example, prudential services are provided for important data; minimum delay is enabled for time-sensitive multimedia and voice applications.
- In addition, QoS lays a good foundation for future integration.

QoS under Differentiated Services

- What are differentiated services

For Differentiated Services (DiffServ), several services with the same feature are converged by using the mechanisms of DSCP (DiffServ CodePoint) and PHB to provide services for the entire converged traffic, instead of individual services.

The following three levels of differentiated services are provided:

1. Expedited Forwarding (EF)
2. Assured Forwarding (AF)
3. Best Effort (BE)

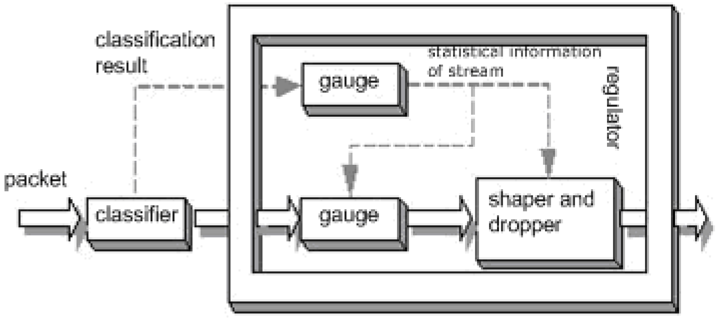
DiffServ greatly reduces the work of signaling and focuses instead on flow convergence and a set of “hop-by-hop behaviors” suitable for networkwide services. Data flows can be classified according to the pre-determined rules in order to converge multiple application traffic into a certain level of data flow.

Therefore, QoS is based on DiffServ system.

- QoS under Differentiated Services

In DiffServ, QoS includes classification/identification, measurement/shaping/packet dropping, and finally congestion management and congestion avoidance. The basic processing is illustrated as below:

Figure 1 Schematic diagram for the border node to classify and adjust the packets



The follow-up chapters separately describe the QoS under differentiated services.

IPv6 QoS

IPv6 QoS supports the 8-bit Traffic Class field, which has the same function as the ToS field of IPv4 to identify the service type of packets. The DSCP of IPv6 takes the first 6 bits in the Traffic Class filed as DHCP value. Thus DSCP=(TC&11111100)>>2. TC value is given by DSCP-to-TC mapping. IPv6 QoS also supports the 20-bit flow label field indicating the traffic belonging to the same classification. Currently, the definition and use of flow label is still in the draft phase, only Traffic Class-based QoS is supported by IPv6 QoS.

Congestion Management

What Is Congestion Management

Congestion occurs in a network node when packets arrive faster than an interface can send them. If the buffer space is insufficient for storing data on the network node, packet loss occurs. Network protocols, such as the Transmission Control Protocol (TCP), provide the data retransmission mechanism, in which the data sender node retransmits data when failing to receive a reply from the peer receiver node. This causes congestion on the peer receiver node and the performance of the entire network deteriorates.

The following are sample causes of congestion:
A packet flow enters **the router** through a high-speed link and leaves the device through a low-speed link;
Packet flows enter **the router** concurrently through multiple interfaces and leaves the device through only one interface;
CPU runs slowly.

Assume that the network needs to transmit some important data as well as much unimportant data. If the device processes all data in the same way with no regard to their importance level, unimportant data uses bulk network bandwidth and transmission of important data is delayed, which may cause tremendous loss.

To resolve these problems, congestion management is introduced. Congestion management features allow you to control congestion by determining the order in which packets are sent out an interface based on priorities assigned to those packets. A network device such as router determines the packet transmission order by controlling which packets are transmitted with priority, ensuring that key services are processed timely.

Policies of Congestion Management

Congestion management performs three tasks:
Create various types of queues.
Classify packets and place them to different queues.
Schedule queues and send packets in queues based on rules. The congestion management QoS features provide five types of queuing. Each type of queuing allows you to create a different number of queues. During periods with light traffic, that is, when no congestion exists, packets are sent out of the interface as soon as they arrive. During periods of transmit congestion at the outgoing interface, packets arrive faster than the interface can send them. If you use congestion management features, packets accumulating at an interface are queued until the interface is free to send them; they are then scheduled for transmission according to their assigned priority and the queuing mechanism configured for the interface. The device determines the packet transmission order by controlling which packets are placed in which queue and how queues are serviced with respect to each other.

This document discusses six types of queuing, which constitute the congestion management QoS features.

Note The NPE80 supports only the First-In, First-Out Queuing (FIFO) and Weighted Fair Queuing (WFQ).

FIFO

FIFO entails no concept of priority or traffic classes. With FIFO, transmission of packets out of the interface occurs in the order the packets arrive. FIFO is the default queuing mechanism that needs no intentional configuration.

WFQ

WFQ offers dynamic, fair queuing that divides bandwidth across queues of traffic based on weights. WFQ ensures that all traffic is treated fairly, given its weight. Given this handling, WFQ ensures satisfactory response time to critical applications, such as interactive, transaction-based applications, which are intolerant of performance degradation.

For WFQ, you define traffic classes based on source addresses, destination addresses, source port numbers, destination port numbers, and protocol types.

CBWFQ

Class-Based Weighted Fair Queuing (CBWFQ) extends the standard WFQ functionality. Same as WFQ, CBWFQ offers dynamic, fair queuing that divides bandwidth across queues of traffic based on weights. The difference lies in classification rules and weight calculation. For WFQ, you define traffic classes based on source addresses, destination addresses, source port numbers, destination port numbers, and protocol types. For CBWFQ, you define traffic classes based on user-defined criteria. WFQ weighs packet priorities based on fixed rules, for example, weighing the priority of an IP packet based on the Type of Service (ToS) domain. CBWFQ assigns communication queue bandwidth by proportion by weighing priorities based on user-defined bandwidth criteria.

CBWFQ allows you to define traffic classes and assign bandwidth in real time. Given these features, CBWFQ allows you to customize bandwidth assignment and ensures that different types of network data traffic acquire bandwidth by proportion.

PQ

With PQ, packets belonging to one priority class of traffic are sent before all lower priority traffic to ensure timely delivery of those packets.

PQ guarantees strict priority for important network data and quickest processing of most-important network data on a network node where PQ is enabled. For PQ, priority can be defined flexibly based on network protocols such as IP, data input interfaces, packet lengths, source addresses, and destination addresses.

LLQ and RTPQ

LLQ indicates Low Latency Queuing. The LLQ feature brings strict Priority Queuing (PQ) to CBWFQ. Strict PQ allows packets that are delay-sensitive and match traffic criteria to be sent and sent before packets in CBWFQ queues.

Functions of Real-time Transport Protocol Priority Queuing (RTPQ) are similar to those of LLQ. Each interface possesses an RTPQ queue exclusively used in the low-delay transmission of RTP packets. RTPQ matches the User Datagram Protocol (UDP) packets of ports in a specified range.

CQ

With Custom Queuing (CQ), each class of traffic acquires bandwidth by proportion. CQ allows you to assign bandwidth to all classes of traffic based on packet importance. Thus, important packets are delivered timely. You can also define the number of the bytes or packets abstracted from queues.

This function is suitable for interfaces that process data at low speeds.

Note You can assign only one queuing mechanism to an interface.

Deciding Which Queuing Policy to Use

Ruijie devices can meet service quality requirements of different services to some degree by implementing FIFO, PQ, CQ, WFQ, CBWFQ, and LLQ&RTPQ. The following section compares the five main queuing strategies.

- FIFO queuing performs the default first-come-first-serviced priorization of packets in user data traffic. It entails no concept of priority or classes of traffic. When FIFO is used, ill-behaved sources can consume available bandwidth, bursty data sources can cause delay in time-sensitive or important traffic, and important traffic may be dropped because less important traffic fills the queue.

- CQ guarantees some level of service to all traffic because you can assign bandwidth to all classes of traffic. You can define the size of the queue by determining its configured packet-count capacity, thereby controlling bandwidth usage.
- PQ guarantees strict priority in that it ensures that one type of traffic will be sent, possibly at the expense of all others. For PQ, a low priority queue can be detrimentally affected, and, in the worst case, never allowed to send its packets if a limited amount of bandwidth is available or if the transmission rate of critical traffic is high.
- WFQ does not use access lists to determine the preferred traffic on an interface. Rather, the fair queue algorithm dynamically sorts traffic into messages that are part of a conversation. Low-volume, interactive traffic gets fair allocation of bandwidth with WFQ, as does high-volume traffic such as file transfers.
- CBWFQ allows you to define traffic classes and assign bandwidth in real time. Specifically, CBWFQ allows you to specify the exact amount of bandwidth to be assigned for a specific class of traffic, which guarantees bandwidth of certain network applications. You can control bandwidth allocation on demand at any time.

The following table compares the main queuing strategies.

	FIFO	**WFQ**	**CBWFQ**	**PQ**	**CQ**
Number of Queues	1	Configurable (default: 256)	320 (64 CBWFQ queues, and 256 WFQ queues)	4	17 (16 user queues, and one system queue)
Advantage	Fast and simple processing	All traffic is treated fairly, given its weight.	Traffic that matches user-defined criteria is placed in CBWFQ queues, and other traffic is placed in WFQ queues. Network packets are delivered in proportion to their configured bandwidth.	High priority queues are serviced first. Absolute prioritization ensures critical traffic of highest priority.	CQ assigns bandwidth by proportion for different types of services. If one queue is empty, its assigned bandwidth is automatically added to another queue that had packets ready to send.
Defect	All packets are treated fairly. Packet delivery sequence is determined by their sequence of arriving at the interface, which may cause delay in delivery of critical applications.	Processing speed is slower than that of FIFO.	Processing speed is slower than that of FIFO.	Processing speed is slow, and a low priority queue can be never allowed to send its packets in the worst case.	Processing speed is relatively slow.
Configuration requirement	No configuration required	Simple configuration required	Configuration required	Configuration required	Configuration required

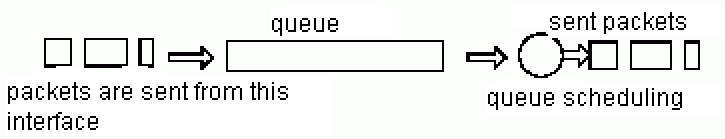
Work Mechanism of Congestion Management

FIFO

Working Principles

FIFO implements a simple service rule and provides only one queue. Packets are delivered based on the first-come-first serviced rule. FIFO does not guarantee delay restrictions or delivery rates, cannot isolate service flows that share a link. Therefore, FIFO cannot fairly treat service flows that share a link.

Figure 2 FIFO queuing



As shown in the preceding figure, packets are sent out an interface in the order in which they arrive. FIFO does not intervene with packets, but allows packets to use bandwidth and other resources in the order in which they arrive. Ill-behaved applications or attacks can consume all the bandwidth and important traffic can be dropped.

Applicable Environment

When there is no other queuing mechanism configured on the network, FIFO is enabled on all interfaces by default. FIFO possesses the quickest processing speed among queuing mechanisms. If congestion rarely occurs, FIFO is the most suitable on the network.

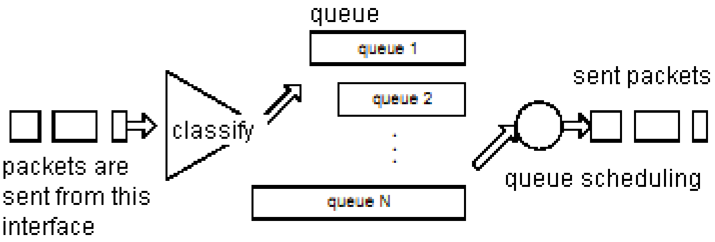
WFQ

Working Principles

WFQ removes FIFO restrictions. When FIFO queuing is enabled, traffic is sent out an interface in the order in which they arrive, disregarding bandwidth consumption and delay. As a result, file transfer and other network applications with large data volumes often generate packet links. A packet link consists of groups of packets and is transferred as a whole. These packet links may consume all available bandwidth and other traffic is dropped.

WFQ ensures that each flow shares link bandwidth fairly and that low-volume traffic gets transmitted in a timely fashion.

Figure 3 WFQ



As shown in the preceding figure, WFQ classifies packets by flow. A flow consists of packets of same source IP address, destination IP address, source MAC address, destination MAC address, source port number, destination port number, protocol type, and ToS. A flow is assigned to a queue with bandwidth in proportion to flow priority during delivery. A flow of high-priority gets more bandwidth than a flow of low-priority. Specifically, bandwidth assigned to a flow divided by the entire bandwidth of the link is equal to the sum (flow priority + 1) divided by the entire bandwidth of the link.

For example, seven flows exist on an interface, with their priority being 1, 2, 3, 4, 5, and 6. The bandwidth assigned to the interface equals the sum of all the flow priorities plus one:

1 + 2 + 3 + 4 + 5 + 6 + 7 = 28

Bandwidth ratio of a flow equals to the sum (flow priority + 1) divided by another sum (sum of all flow priorities + 1). Therefore, bandwidth ratios of flows are 1/28, 2/28, 3/28, 4/28, 5/28, 6/28, and 7/28.

If there are 10 data flows whose priority is 1, the entire bandwidth is:

1 + 2*10 + 3 + 4 + 5 + 6 + 7 = 46

Priority-0 data flows use 1/46 of the entire bandwidth, Priority-1 data flows use 2/46 of entire bandwidth, and accordingly Priority-6 data flows use 7/46 of the entire bandwidth.

When data flows are added or terminated, actual bandwidth changes consequently. By assigning bandwidth of each flow in real time, WFQ is suitable for the network environment that is constantly changing.

Applicable Environment

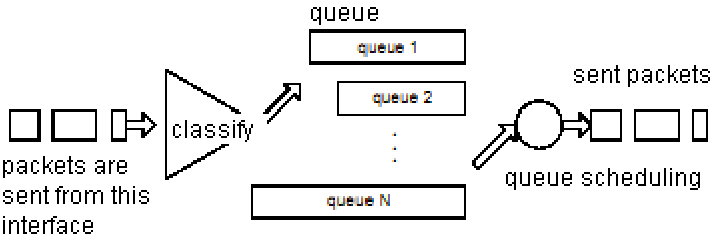
WFQ is suitable for serial interfaces with bandwidth equal to or less than 2.048 Mbps.

CBWFQ

Working Principles

CBWFQ extends the standard WFQ functionality to provide support for assigning bandwidth based on user-defined traffic classes.

Figure 4 CBWFQ



As shown in the preceding figure, CBWFQ defines classes for packets by flow. At first, classes are defined based on user-defined based on user-defined rules. Packets satisfying a same match criterion belong to a network data flow and are placed in a same CBWFQ queue. Packets satisfying no match criterion are classified in WFQ mode. Packets with the same source IP address, destination IP address, source MAC address, source port number, destination port number, protocol type, and ToS field belong to the same flow. When a packet is assigned to a flow, it is placed in the WFQ queue reserved for that flow. During delivery, CBWFQ assigns bandwidth based on the user-defined rules.

When data flows are added or terminated, actual bandwidth assigned by CBWFQ changes consequently. Therefore, CBWFQ is also suitable for the network environment that is constantly changing.

Applicable Environment

CBWFQ is suitable for applying strict classification rules and bandwidth allocation on serial interfaces with bandwidth equal to or less than 2.048 Mbps.

LLQ and RTPQ

Working Principles

LLQ extends the standard CBWFQ functionality. LLQ ensures that delay-sensitive data acquire bandwidth and are sent before packets in other queues are dequeued. The LLQ feature brings strict Priority Queuing (PQ) to CBWFQ. Strict PQ allows packets in LLQ queues to be sent and sent before packets in CBWFQ queues.

For LLQ queues, traffic of different types is monitored separately. If there is no congestion, traffic is allowed for delivery. In the event of congestion, transmission rates of all traffic are monitored, and packets are dropped if the bandwidth is exceeded.

Functions of RTPQ are similar to those of LLQ. Each interface possesses an RTPQ queue exclusively used in the low-delay transmission of RTP packets. RTPQ matches the UDP packets of ports in a specified range.

For RTPQ queues, traffic of different types is monitored separately. If there is no congestion, traffic is allowed for delivery. In the event of congestion, transmission rates of all traffic are monitored, and packets are dropped if the bandwidth is exceeded.

Both an RTPQ queue and an LLQ queue belongs can belong to only one interface, but the priority of an RTPQ queue is higher than that of an LLQ queue.

Applicable Environment

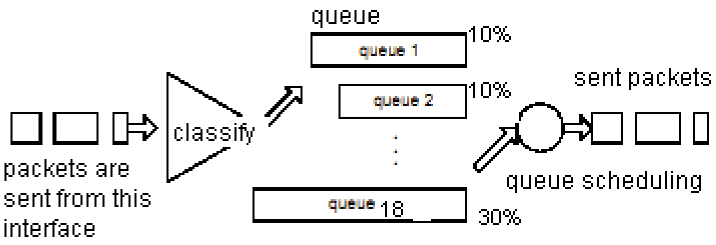
LLQ and RTPQ+CBWFQ are suitable for applying strict classification rules and bandwidth allocation on serial interfaces with bandwidth equal to or less than 2.048 Mbps.

CQ

Working Principles

CQ defines 17 classes for packets, corresponding to 17 CQ queues. Packets enter corresponding CQ queues based on their classes and the FIFO policy. For queues 1 through 16, queue 0 is a system queue; queue 1 through 16 are user queues. Associated with each queue is a configurable byte count, which specifies how many bytes of data the system should deliver from the current queue before it moves on to the next queue. The system queue is emptied before any of the queues 1 through 16 are processed. For queues 1 through 16, the system cycles through the queues using the pre-assigned bandwidth in a round-robin fashion, the system dequeues the configured byte count from each queue in each cycle and delivers packets in the current queue before moving on to the next one.

Figure 5 CQ



CQ ensures that no application or specified group of applications achieves more than a predetermined proportion of overall bandwidth when the line is under stress. Like PQ, CQ is statically configured and does not automatically adapt to changing network conditions. With CQ enabled, the system takes longer and consumes more resources to switch packets because packets need to be queued.

How to Determine Byte Count Values for Queues

In order to assign bandwidth to different queues, you must specify the byte count for each queue. This section describes how to determine byte count values for queues.

When the router cycles through queues in round-robin fashion, the device sends packets from a particular queue until the byte count is exceeded. If the byte count value is exceeded but packets of the queue are not completely sent, the device continues sending until all packets of the queue are sent. Therefore, if you set the byte count to 300 bytes and the packet size of your protocol is 1500 bytes, then every time this queue is serviced, 1500 bytes will be sent, not 300 bytes.

For example, suppose one protocol has 500-byte packets, another has 300-byte packets, and a third has 100-byte packets. If you want to split the bandwidth evenly across all three protocols, you might choose to specify byte counts of 200, 200, and 200 for each queue. However, this configuration does not result in a 33/33/33 ratio. When the device services the first queue, it sends a single 500-byte packet; when it services the second queue, it sends a 300-byte packet; and when it services the third queue, it sends two 100-byte packets. The effective ratio is 50/30/20.

Thus, setting the byte count too low can result in an unintended bandwidth allocation. However, very large byte counts will produce a “jerky” distribution. That is, if you assign 10 KB, 10 KB, and 10 KB to three queues in the example given, each protocol is serviced promptly with equal bandwidth assigned when its queue is the one being serviced, but it may be a long time before the queue is serviced again. A better solution is to specify 500-byte, 600-byte, and 500-byte counts for the queue. This configuration results in a ratio of 31/38/31, which may be acceptable. In order to service queues in a timely manner and ensure that the configured bandwidth allocation is as close as possible to the required bandwidth allocation, you must determine the byte count based on the packet size of each protocol; otherwise your percentages may not match what you configure.

To determine the correct byte counts, perform the following steps:

1. For each queue, divide the percentage of bandwidth you want to assign to the queue by the packet size, in bytes. For example, assume the packet size for protocol A is 1086 bytes, protocol B is 291 bytes, and protocol C is 831 bytes. You want to assign 20 percent for A, 60 percent for B, and 20 percent for C. The ratios would be:
   20/1086, 60/291, 20/831
   or 0.01842, 0.20619, 0.02407

2. Normalize the numbers by dividing by the lowest number:
   1, 11.2, 1.3
   The result is the ratio of the number of packets that must be sent so that the percentage of bandwidth that each protocol uses is approximately 20, 60, and 20 percent.

3. A fraction in any of the ratio values means that an additional packet will be sent. Round up the numbers to the next whole number to obtain the actual packet count. In this example, the actual ratio will be 1 packet, 12 packets, and 2 packets.

4. Convert the packet number ratio into byte counts by multiplying each packet count by the corresponding packet size. In this example, the number of packets sent is one 1086-byte packet, twelve 291-byte packets, and two831-byte packets, or 1086, 3492, and 1662 bytes, respectively, from each queue. These are the byte counts you would specify in your CQ configuration.

5. To determine the bandwidth distribution this ratio represents, first determine the total number of bytes sent after all three queues are serviced:
   (1 x 1086) + (12 x 291) + (2 x 831) = 1086 + 3492 + 1662 = 6240

6. Then determine the percentage of the total number of bytes sent from each queue:
   1086/6240, 3492/6240, 1662/6240 = 17.4, 56, and 26.6 percent
   This result is close to the desired ratio of 20/60/20.

7. If the actual bandwidth is not close enough to the desired bandwidth, multiply the original ratio of1:11.2:1.3 by the best value, trying to get as close to three integer values as possible. Note that the multiplier you use need not be an integer. For example, if we multiply the ratio by two, we get 2:22.4:2.6. You would now send two 1086-byte packets, twenty-three 291-byte packets, and three 831-byte packets, or 2172/6693/2493, for a total of 11,358 bytes. The resulting ratio is 19/59/22 percent, which is much closer to the desired ratio that we achieved.

Window size also affects the bandwidth distribution. If the window size of a particular protocol is set to one, then that protocol will not place another packet into the queue until it receives an acknowledgment. The CQ algorithm moves to the next queue if the byte count is exceeded or no packet is in that queue. Therefore, with a window size of one, only one frame will be sent each time. If your frame count is set to 2 kilobytes, and your frame size is 256 bytes, then only 256 bytes will be sent each time this queue is serviced.

Applicable Environment

This function is suitable for interfaces that process data at low speeds.

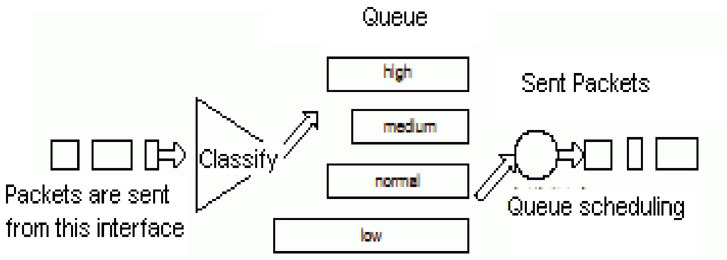
Note CQ is not supported on any tunnels.

PQ

Working Principles

PQ allows you to define how traffic is prioritized in the network. You can configure four traffic priorities. You can define a series of filters based on packet characteristics (source addresses, destination addresses, protocol types, and packet sizes) to place traffic into these four queues; the queue with the highest priority is serviced first until it is empty, then the lower queues are serviced in sequence.

Figure 6 PQ



As shown in the preceding figure, packets are classified and placed in four output types of output queues based on user-defined criteria. The priority queues on that interface are scanned for packets in descending order of priority. The high priority queue is scanned first, then the medium priority queue, and so on. The packet at the head of the highest queue is chosen for transmission. This procedure is repeated every time a packet is to be sent.

When choosing to use PQ, consider that because lower priority traffic is often denied bandwidth in favor of higher priority traffic, use of PQ could, in the worst case, result in lower priority traffic never being sent. PQ introduces extra system resource consumption that is acceptable for slow interfaces, but may not be acceptable for higher speed interfaces such as Ethernet. With PQ enabled, the system takes longer to switch packets, degrading system performance. PQ uses a static configuration and does not adapt to changing network conditions.

Applicable Environment

Although you can enable PQ for any interface, it is best used for low-bandwidth, congested serial interfaces.

Note PQ is not supported on any tunnels.

WFQ

WFQ Configuration Tasks

When standard WFQ is enabled, packets are classified by flow. Packets with the same source IP address, destination IP address, source TCP or UDP port, and destination TCP or UDP port belong to the same flow. WFQ assigns an equal share of bandwidth to each flow. Flow-based WFQ is also called fair queueing because all flows are equally weighted.

Configuring WFQ entails the following two processes:

- Configuring WFQ
- Monitoring fair queuing

Note Memory configuration is different for routers on different platforms. You are advised to run the **fair-queue** command to configure depth and count of different queues.

Configuring WFQ

To configure WFQ, run the following commands in the interface configuration mode:

Command	Function
Ruijie(config-if)# **fair-queue** [*congestive-discard-threshold* [*dynamic-queues*]]	Configure WFQ.
Ruijie(config-if)# **no fair-queue**	Cancel WFQ configuration.

The following table describes parameters related to WFQ configuration.

Command	Function
congestive-discard-threshold	Specify the maximum threshold of packets allowed in each queue. The default value is 64 and the value range is 1–4096. When the packet count reaches the threshold, newly arrived packets will be discarded. This parameter is optional.
dynamic-queues	Specify the number of dynamic queues. The default value is 256 and the value must be 2^n within 16–4096. This parameter is optional.

Note To configure the WFQ congestion management policy, ensure that the fast-switching function configuration (enable or disable the function) is consistent on all system interfaces. Otherwise, the congestion management policy becomes invalid.

Note The value of *dynamic-queues* needs to be adjusted dynamically according to the current service traffic status. Ensure that the value of *dynamic-queues* be greater than the number of service flows; otherwise, multiple service flows enter a same dynamic queue. You are advised to set the *dynamic-queues* to a value greater than both 64 and the number of service flows.

Monitoring WFQ

To view information about WFQ that has been configured, run the following commands in the privileged user mode:

Command	Function

Ruijie# **show queue wfq**	Show configuration information about WFQ queues.
Ruijie# **show queue interface** *interface-name interface-number*	Show interface statistic information about WFQ queues.

For routers of the RSR series, you can view statistic information about fast-switching WFQ queue interfaces by running the **show queue interface** command. "Qos Ref queue information" identifies statistic information about fast-switching WFQ queue interfaces.

WFQ Configuration Example

In the following instance, fair queueing is configured on synchronization port 0. Specifically, serial port 0 is configured with 128 message queues, 512 dynamic queues, and 50 reserved queues.

```
interface Serial 1/0
ip address 1.1.1.1 255.255.255.0
fair-queue 128 512
```

The following are examples of viewing the interface configuration in the privileged user mode:

```
Ruijie# show queue interface serial 1/0
Queuing strategy: weighted fair
Output queue: 0/300/128/0 (size/max total/threshold/drops)
Output queue num: 0/0/512 (now active/max active/max total)
```

The preceding output shows that WFQ is enabled on the interface and the threshold for discarding packets is **128** in the event of congestion.

CBWFQ

CBWFQ Configuration Tasks

By default, the QoS policy of FIFO is enabled on Ruijie network interfaces with bandwidth lower than 4 Mbps.

Configuring CBWFQ

Defining Class-Maps

This function is required for realizing CBWFQ functionality. You can define packet classification rules in class-maps, and specify the class-map names for using them in policy-maps. A class-map can be used by one or more policy-maps. The following table describes the typical configuration of this function.

Command	**Function**
Ruijie(config)# **class-map** *match-all class-map-name*	Access and create the class-map of the "AND" type. That is, all conditions in the class-map must be met.
Ruijie(config)# **class-map** *match-any class-map-name*	Access and create the class-map of the "OR" type. That is, all conditions in the class-map must be met.
Ruijie(config-cmap)# **match access-group** *access-list-number* or Ruijie(config-cmap)# **match input-interface** *interface-name* or Ruijie(config-cmap)# **match protocol** *protocol-name* or Ruijie(config-cmap)# **match cos** *value* or Ruijie(config-cmap)# **Match ip dscp** *value* or Ruijie(config-cmap)# **Match ip precedence** *value* or Ruijie(config-cmap)# **Match not** *match-type value*	Configure the packet classification rules. The rules can be the positive or negative forms of the ACL, packet receiving interface, encapsulation protocol type, COS value, IP DSCP code, and IP Precedence code.
Ruijie(config-cmap)# **exit**	Exit from the configuration layer of class-maps.

Class-map-name: indicates the name of a class-map.

Match-all: All conditions in the class-map must be met. By default, a created class-map is of the **Match-all** type.

Match-any: indicates that only one condition in the class-map needs to be met.

Access- list-number: indicates the number of the access control list (ACL).

Interface- name: indicates the name of a network interface.

Protocol-name: indicates the packet encapsulation protocol.

COS: indicates the COS value of an Ethernet packet.

IP dscp: indicates the DSCP code value of the packet IP TOS domain.

IP Precedence: indicates the Precedence code value of the packet IP TOS domain.

Not match-type: indicates the negative form of a classification rule.

Configuring Rules in Policy-Maps

This function is required for realizing CBWFQ functionality. In a policy-map, you can use all class-maps configured on the router and use a maximum of 64 class-maps concurrently. You can define bandwidth for class-maps that are used under the precondition that the sum of bandwidth for all class-maps is equal to or smaller than the bandwidth assigned to CBWFQ on a specified interface that applies the policy-map. The following table describes the typical configuration of this function.

Command	Function
Ruijie(config)# **policy-map** *policy-map-name*	Access and create a policy-map.
Ruijie(config-pmap)# **class** *class-map-name*	Use class-maps that have been defined.
Ruijie(config-pmap-c)# **bandwidth** { *bandwidth-kbps* | **percent** *percent-number* }	Specify bandwidth assigned to specified types of data flows.
Ruijie(config-pmap-c)# **queue-limit** *number-of-packets*	Set the queue depth.
Ruijie(config-pmap-c)# **exit**	Exit from the configuration layer of class using.
Ruijie(config-pmap)# **exit**	Exit from the configuration layer of policy-maps.

Policy-map-name: indicates the name of a policy-map.

Class-map-name: indicates the name of a class-map.

Bandwidth-kbps: indicates the assigned bandwidth in kbps.

Percent-number: indicates the assigned-bandwidth to all-available-bandwidth ratio.

Number-of-packets: indicates the depth of a CBWFQ queue (the maximum number of packets that are allowed).

Applying Service Rules on Specified Interfaces

This function is required for realizing CBWFQ functionality. If service rules are enabled on specified interfaces, CBFWQ functionality is enabled and classes used in policy-maps obtain their processing queues. The following table describes the typical configuration of this function.

Command	Function
Ruijie(config-if)# **service-policy output** policy-map-name	Enable CBWFQ and specify the policy-maps to be applied.
Ruijie(config-if)# **service-policy input** policy-map-name	Enable the policy-map strategy for interface input packets.

Policy-map-name: indicates the name of a policy-map.

Note

If CBWFQ characteristics (such as the **Bandwidth** command or the **Priority** command) are enabled for a policy-map, the policy-map can be only be used in the **Service-policy Output** command, but not the **Service-policy Input** command.

Note

To configure the policy-map of an interface, ensure that the fast-switching function configuration (enable or disable the function) is consistent on all system interfaces. Otherwise, the functions such as CBWFQ and police of the policy-map become invalid.

Note

Ensure that the fast-switching function is disabled, if an interface configured with an input-direction policy-map or associated with an output-direction policy mapping of the shape and red functions. This is a required in the current software version.

Configuring Bandwidth for Specified Classes

This function is optional for realizing CBWFQ functionality. The following table describes the typical configuration of this function.

Command	Function
Ruijie(config)# **policy-map** *policy-map-name*	Access and create a policy-map.
Ruijie(config-pmap)# **class** *class-name*	Use class-maps that have been defined.
Ruijie (config-pmap-c)# **bandwidth** { bandwidth-kbps | **percent** percent}	Specify bandwidth assigned to specified types of data flows.

Policy-map-name: indicates the name of a policy-map.

Class-map-name: indicates the name of a class-map.

Bandwidth-kbps: indicates the assigned bandwidth in kbps.

Percent-number: indicates the assigned-bandwidth to all-available-bandwidth ratio.

You can specify bandwidth assigned to specified types of data flows. By default, the system assigns 1% of the bandwidth to a specified type of data flow.

Configuring Queue Depth for Specified Classes

This function is optional for realizing CBWFQ functionality. The following table describes the typical configuration of this function.

Command	Function
Ruijie(config)# **policy-map** *policy-map-name*	Access and create a policy-map.
Ruijie(config-pmap)# **class** *class-name*	Use class-maps that have been defined.
Ruijie(config-pmap-c)# **queue-limit** *number-of-packets*	Set the queue depth.

Policy-map-name: indicates the name of a policy-map.

Class-map-name: indicates the name of a class-map.

Number-of-packets: indicates the depth of a CBWFQ queue (the maximum number of packets that are allowed).

You can set the depth of a CBWFQ queue corresponding to a specified type of data flow. The system default value is **64**. That is, the system will discard packets attempting to enter a CBFWQ queue if the queue has got 64 packets. In this case, Ruijie products support the congestion processing only in Tail-Drop mode, not in WRED mode.

Note

Queue depth configuration depends on network requirements. When forwarded data is delay-sensitive, you can decrease the queue depth to lower delay. When the forwarded data is burst or contains many small packets, you can increase the queue depth to improve the system buffer capability.
Do not adjust the queue depth to a value that is too small; otherwise, the bandwidth guarantee function

becomes abnormal.

In an environment where the forwarded data is burst or contains many small packets, bandwidth may fail to be guaranteed. You need to increase the queue depth to improve the system buffer capability.

Configuring the DSCP Code Value of the IP TOS Domain for a Specified Class

This function is optional for realizing CBWFQ functionality. The following table describes the typical configuration of this function.

Command	Function
Ruijie(config)# **policy-map** *policy-map-name*	Access and create a policy-map.
Ruijie(config-pmap)# **class** *class-name*	Use class-maps that have been defined.
Ruijie(config-pmap-c)# **set ip dscp** *values*	Set the packet DSCP value.

Policy-map-name: indicates the name of a policy-map.

Class-map-name: indicates the name of a class-map.

Values: indicates the packet DSCP value to be set.

Configuring the DSCP Code Value of the IPv4 TOS Domain and the IPv6 Traffic Class Domain for a Specified Class

This function is optional for realizing CBWFQ functionality. The following table describes the typical configuration of this function.

Command	Function
Ruijie(config)# **policy-map** *policy-map-name*	Access and create a policy-map.
Ruijie(config-pmap)# **class** *class-name*	Use class-maps that have been defined.
Ruijie(config-pmap-c)# **set dscp** *values*	Set the packet DSCP value.

Policy-map-name: indicates the name of a policy-map.

Class-map-name: indicates the name of a class-map.

Values: indicates the packet DSCP value to be set.

Configuring the Precedence Code Value of the IP TOS Domain for a Specified Class

This function is optional for realizing CBWFQ functionality. The following table describes the typical configuration of this function.

Command	Function
Ruijie(config)# **policy-map** *policy-map-name*	Access and create a policy-map.
Ruijie(config-pmap)# **class** *class-name*	Use class-maps that have been defined.
Ruijie(config-pmap-c)# **set ip precedence** *values*	Set the packet Precedence value.

Policy-map-name: indicates the name of a policy-map.

Class-map-name: indicates the name of a class-map.

Values: indicates the packet Precedence value to be set.

Configuring the Precedence Code Value of the IPv4 TOS Domain and the IPv6 Traffic Class Domain for a Specified Class

This function is optional for realizing CBWFQ functionality. The following table describes the typical configuration of this function.

Command	Function

Ruijie(config)# **policy-map** *policy-map-name*	Access and create a policy-map.
Ruijie(config-pmap)# **class** *class-name*	Use class-maps that have been defined.
Ruijie(config-pmap-c)# **set precedence** *values*	Set the packet Precedence value.

Policy-map-name: indicates the name of a policy-map.

Class-map-name: indicates the name of a class-map.

Values: indicates the packet Precedence value to be set.

Configuring the COS Value of Ethernet Packets for a Specified Class

This function is optional for realizing CBWFQ functionality. The following table describes the typical configuration of this function.

Command	**Function**
Ruijie(config)# **policy-map** *policy-map-name*	Access and create a policy-map.
Ruijie(config-pmap)# **class** *class-name*	Use class-maps that have been defined.
Ruijie(config-pmap-c)# **set cos** *values*	Set the packet COS value.

Policy-map-name: indicates the name of a policy-map.

Class-map-name: indicates the name of a class-map.

Values: indicates the packet COS value to be set.

Configuring Bandwidth Assigned to CBWFQ

This function is optional for realizing CBWFQ functionality. The following table describes the typical configuration of this function.

Command	Function
Ruijie(config-if)# **max-reserved-bandwidth** percent	Set bandwidth assigned to CBWFQ.

Percent: indicates percentage of the bandwidth assigned to CBWFQ on an interface.

You can customize the value of the **percent** parameter. The system default value is **75**. That is, 75% of all available bandwidth of the corresponding network interface will be assigned to CBWFQ queues.

Monitoring CBWFQ

To view information about input and output queues when CBWFQ is enabled on an interface, run the following commands in the privileged user mode:

Command	**Function**
Ruijie# **show class-map**	Show information about all class-maps.
Ruijie# **show class-map** *class-map-name*	Show information about a specified class-map.
Ruijie# show policy-map	Show information about all policy-maps.
Ruijie# **show policy-map name** *policy-map –name*	Show information about a specified policy-map.
Ruijie# **show policy-map name** *policy-map -name* **class** *class-name*	Show information about specified class-maps in a specified policy-map.
Ruijie# **show policy-map interface** interface-name interface-number	Show information about policy-maps applied on a specified interface.
Ruijie# **show queue interface** interface-name *interface-number*	Show interface statistic information about CBWFQ queues.

Class-map-name: indicates the name of a class-map.

Policy-map-name: indicates the name of a policy-map.

Interface-name: indicates the name of a network interface.

Note

For routers of the RSR series, you can view statistic information about fast-switching CBWFQ queue interfaces by running the **show queue interface** command. "Qos Ref queue information" identifies statistic information about fast-switching CBWFQ queue interfaces.

Note

You can run the **show policy-map interface** command to view statistic information about fast-switching policy-maps on routers of the RSR series.

CBWFQ Configuration Example

In the following example, CBWFQ is enabled on a WAN interface Serial1/0 (S1/0) of the router. 30% of all available bandwidth on the interface S0 is used in the IP communication between the host 192.168.201.213 and the host 192.168.12.216. 10% of all available bandwidth on the interface S0 is used in the IP communication between the host 192.168.201.213 and the host 192.168.12.77.

The following is the configuration of the device where CBWFQ is enabled:

```
Ruijie# show running-config
!
.. .. ..
!
class-map class1
match access-group 101
class-map class2
match access-group 102
!
policy-map policy1
class class1
bandwidth percent 30
class class2
bandwidth percent 10
!
interface FastEthernet0/0
ip address 192.168.201.1 255.255.255.0
!
interface Serial1/0
ip address 192.168.20.3 255.255.255.0
encapsulation ppp
service-policy output policy1
clock rate 115200
!
ip route 0.0.0.0 0.0.0.0 Serial1/0
access-list 101 permit ip host 192.168.201.213 host 192.168.12.216
access-list 102 permit ip host 192.168.201.213 host 192.168.12.77
!
.. .. ..
```

In the following example, the DSCP value is set for five types of packets matching the ACL on the ingress interface and the packets are transformed into four types of macro-flows for the differentiated services. The bandwidth assigned to CBWFQ queues is adjusted.

The following is the configuration of the device where CBWFQ is enabled:

```
class-map match-all dlsw
match access-group 101
class-map match-all voip
match access-group 102
class-map match-all notes
match access-group 103
class-map match-all http
match access-group 104
class-map match-all ftp
match access-group 105
class-map match-all ef
match ip dscp 46
```

```
class-map match-all af1
match ip dscp 10
class-map match-all af2
match ip dscp 18
class-map match-all af3
match ip dscp 26
!
!
policy-map 1
class dlsw
set ip dscp 46
class voip
set ip dscp 46
class notes
set ip dscp 10
class http
set ip dscp 18
class ftp
set ip dscp 26
policy-map 2
class ef
bandwidth 800
class af1
bandwidth 500
class af2
bandwidth 300
class af3
bandwidth 380
!
access-list 101 permit udp any any eq 2065
access-list 102 permit udp any any range 16384 32767
access-list 103 permit udp any any eq 1352
access-list 104 permit udp any any eq 80
access-list 105 permit udp any any eq 20
!
interface serial 2/1
encapsulation PPP
no ip route-cache
ip address 1.1.1.2 255.255.255.0
max-reserved-bandwidth 99
service-policy output 2
clock rate 2048000
!
interface FastEthernet 0/0
no ip route-cache
ip address 192.168.200.1 255.255.255.0
service-policy input 1
duplex auto
mac-address 00d0.3456.eeee
speed auto
!
ip route 0.0.0.0 0.0.0.0 serial 2/1
!
```

Configuring LLQ and RTPQ

Configuration Tasks of LLQ and RTPQ

LLQ configuration is combined with CBWFQ configuration, but RTPQ configuration is independent.

Configuring LLQ and RTPQ

To configure LLQ, run the following commands in the configuration mode of the Policy-map command layer:

Command	Function
Ruijie(config)# **policy-map** *policy-map-name*	Access and create a policy-map.

Ruijie(config-pmap)# **class** *class-name*	Use class-maps that have been defined.
Ruijie (config-pmap-c)# **priority** { *bandwidth-kbps* | **percent** *percent*} {*Burst bytes*}	Specify bandwidth assigned to specified types of data flows.
Ruijie(config-if)# service-policy output policy-map-name	Enable CBWFQ and specify the policy-maps to be applied.

Policy-map-name: indicates the name of a policy-map.

Class-map-name: indicates the name of a class-map.

Bandwidth-kbps: indicates the assigned bandwidth in kbps.

Percent-number: indicates the assigned-bandwidth to all-available-bandwidth ratio.

You can specify bandwidth assigned to specified types of data flows. By default, the system assigns 1% of the bandwidth to a specified type of data flow.

Burst bytes: indicates the byte count that can be exceeded.

Note

Queue depth configuration depends on network requirements. When forwarded data is delay-sensitive, you can decrease the queue depth to lower delay. When the forwarded data is burst or contains many small packets, you can increase the queue depth to improve the system buffer capability.
Do not adjust the queue depth to a value that is too small; otherwise, the bandwidth guarantee function becomes abnormal.
In an environment where the forwarded data is burst or contains many small packets, bandwidth may fail to be guaranteed. You need to increase the queue depth to improve the system buffer capability.

To configure RTPQ, run the following commands in the configuration mode of the interface configuration command layer:

Command	Function
Ruijie(config-if)# **ip rtp priority** *starting-rtp-port-number port-number-range bandwidth*	Create RTPQ queues for an interface and assign bandwidth to these queues.

Starting-rtp-port-number: indicates the starting port number among the matching UDP ports.

Port-number-range: indicates the port number range of the matching UDP ports.

Bandwidth-kbps: indicates the assigned bandwidth in kbps.

Note

The following three commands affect the bandwidth assigned for interfaces:
The **Bandwidth** command of CBWFQ
The **Priority** command of LLQ
The **ip rtp priority** command of RTPQ
The **Bandwidth** command of the interface is used to set the entire bandwidth.
That is, Bandwidth = max_reserve +default wfq bandwidth;
max_reserve = bandwidth (policy-map) + priority (policy-map) +ip RTP priority

Note

To configure the interface congestion management policy, ensure that the fast-switching function configuration (enable or disable the function) is consistent on all system interfaces. Otherwise, the congestion management policy becomes invalid.

Monitoring LLQ and RTPQ

To view information about LLQ that has been configured, run the following commands in the privileged user mode:

Command	Function
Ruijie# **show policy-map** *interface-name interface-number*	Show information about policy-maps applied on a specified interface.

To view information about RTPQ that has been configured, run the following commands in the privileged user mode:

Command	Function
Ruijie# **show queue rtpq**	Show information about RTPQ queues.

Note

For routers of the RSR series, you can view statistic information about fast-switching RTPQ queue interfaces by running the **show queue interface** command. "Qos Ref queue information" identifies statistic information about fast-switching RTPQ queue interfaces.

You can run the **show policy-map interface** command to view statistic information about fast-switching policy-maps on routers of the RSR series.

Configuration Examples of LLQ and RTPQ

In the following example, on SYNC port 0, an RTPQ queue is configured for serving RTP of VOIP and an LLQ queue is configured for serving DOSQ office packets. CBFWQ queues process Notes packets and default WFQ queues in CBWFQ process other packets.

```
!
class-map match-all dlsw
match access-group 101
class-map match-all voip
match access-group 102
class-map match-all notes
match access-group 103
class-map match-all http
match access-group 104
class-map match-all ftp
match access-group 105
!
policy-map 3
class dlsw
priority 30 2000
class notes
bandwidth 13
!
access-list 101 permit udp any any eq 2065
access-list 102 permit udp any any range 16384 32767
access-list 103 permit udp any any eq 1352
access-list 104 permit udp any any eq 80
access-list 105 permit udp any any eq 20
!
interface serial 2/1
encapsulation PPP
ip rtp priority 16384 16383 40
no ip route-cache
ip address 1.1.1.2 255.255.255.0
max-reserved-bandwidth 99
service-policy output 3
clock rate 64000
bandwidth 84
!
interface FastEthernet 0/0
ip address 192.168.200.1 255.255.255.0
duplex auto
```

```
mac-address 00d0.3456.eeee
speed auto
!
ip route 0.0.0.0 0.0.0.0 serial 2/1
!
```

Configuring CQ

CQ Configuration Tasks

Configuring CQ

You can configure a maximum of 16 groups for CQ. That is, the List-number range is 1–16. In each group, you can specify queues of packets, queue length, and byte count that can be continuously sent after a queue obtains the controlling right of sending packets.

Determining the Maximum Capacity of CQ

To configure the maximum packet capacity for each queue, run the following commands in the global configuration mode:

Command	Function
Ruijie(config)# **queue-list** *list-number* **queue** *queue-number* limit *limit-number*	Specify the maximum packet count in a CQ queue. The no form of this command can be run to restore the default queue length 20.
Ruijie(config)# **queue-list** *list-number* **queue** *queue-number* **byte-count** *byte-count-number*	Specify the maximum byte count in a queue. The no form of this command can be run to restore the default byte count 1500.

List-number: indicates the number of a queue list within 1–16.

Queue-number: indicates the number of a custom queue number within 1–16.

Limit-number: indicates the maximum packet count in a queue. The value range is 0–32767, and the default value is **20**.

Byte-count-number: specifies how many bytes of data should be delivered from the current queue by the system before the system moves on to the next queue until the value of **Byte-count-number** is exceeded or no packet is in that queue. The value range is 0–16777215, and the default value is **1500**. To determine the correct byte counts, see the related section.

Note

To configure the interface congestion management policy, ensure that the fast-switching function configuration (enable or disable the function) is consistent on all system interfaces. Otherwise, the congestion management policy becomes invalid.

Note

You should configure Byte-count-number based on traffic in each queue. Do not configure a large byte count for a slow traffic queue. Otherwise, the current queue may keep being dispatched, resulting in that other queues cannot be processed.

Assigning Packets to CQ Queues

You can assign a packet to a CQ queue based on the protocol type and the interface where the packet enters the device. You can also configure a default queue with multiple rules for packets matching no allocation criterion.

To configure the CQ queue list, run the following commands in the global configuration mode:

Command	Function
Ruijie(config)# **queue-list** *list-number* **protocol** protocol-name queue-number [queue-keyword] [keyword-value]	Assign a packet to a CQ queue based on the protocol type.
Ruijie(config)# **queue-list** *list-number* **interface** interface-type interface-number queue-number	Assign a packet to a CQ queue based on the type of the interface where the packet enters the device.

Ruijie(config)# **queue-list** *list-number* **default** queue-number	Assign packets matching no assignment rules in the CQ queue list to a CQ queue. The default CQ queue number is **1**.

For the description about **List-number** and **Queue-number**, see the preceding section.

Protocol-name: indicates the type of a protocol. The value can be **IP** (frequently used), **PAD**, and so on.

Queue-keyword and **Keyword-value**: indicates options about protocols. The following table describes values and meanings of the two parameters when IP is used.

queue-keyword	**keyword-value**	**Meaning**
Blank	Blank	All IP packets can enter a specified queue.
fragments	Blank	All fragment IP packets can enter a specified queue.
list	List-number	All packets matching the access-list-number can enter a specified queue.
lt	Byte-count	Packets whose length is smaller than the value of byte-count can enter a specified queue.
gt	Byte-count	Packets whose length is greater than the value of byte-count can enter a specified queue.
tcp	Port	IP packets whose source or destination TCP port number is port can enter a specified queue.
udp	Port	IP packets whose source or destination UDP port number is port can enter a specified queue.

Ensure that the fast-switching function is disabled on an interface when you specify the fragments policy for protocol rules. This is required in the current software version.

Specifying the Lowest CQ Queue Number

By default, you can customize CQ queues numbered 1 to 16. CQ reserves queue 0 for routing protocols and queue 0 functions as the absolute priority queue higher than all customize queues. If there is other traffic that needs to be processed with high priority, CQ allows you to increase the number of absolute priority queues by adjusting the lowest custom queue list number.

By default, the lowest custom queue number is 1. That is, only queue 0 is the absolute priority queue, and queues 1–16 are custom queues.

To configure the lowest CQ queue number, run the following commands in the global configuration mode:

Command	Function
Ruijie(config)# **queue-list** *list-number* **lowest-custom** *queue-number*	Customize the lowest CQ queue number.

List-number: indicates the number of a queue list within 1–16.

Queue-number: indicates the number of a custom queue within 0–16.

Applying the CQ List on an Interface

To apply a CQ list on an interface, run the following command in the interface configuration mode:

Command	Function
Ruijie(config-if)#**custom-queue-list** *list-number*	Set the queueing policy of an interface to a specified CQ list.

Note Each interface can use only one queueing policy and one matching list.

Monitoring CQ

To view information about input and output queues when CQ is enabled on an interface, run the following commands in the privileged user mode:

Command	Function
Ruijie# **show queue cq**	Show configuration information about CQ queues.
Ruijie# **show queue interface** interface-name *interface-number*	Show interface statistic information about CQ queues.
Ruijie# **debug qos cq**	Enable CQ debugging.

Note

For routers of the RSR series, you can view statistic information about fast-switching CQ queue interfaces by running the **show queue interface** command. "Qos Ref queue information" identifies statistic information about fast-switching CQ queue interfaces.

CQ Configuration Examples

Set CQ list 2: Assign IP packets whose length is greater than 200 bytes to queue 12.

```
Ruijie(config)# queue-list 2 protocol ip 12 gt 200
```

Set CQ list 4: Assign IP packets whose length is smaller than 300 bytes to queue 2.

```
Ruijie(config)# queue-list 4 protocol ip 2 lt 300
```

Set CQ list 1: Assign traffic matching IP-access-list 10 to queue 11.

```
Ruijie(config)# queue-list 1 protocol ip 11 list 10
```

Set CQ list 4: Assign Telnet packets to queue 12.

```
Ruijie(config)# queue-list 4 protocol ip 12 tcp 23
```

Set CQ list 4: Assign UDP domain name service packets to queue 2.

```
Ruijie(config)# queue-list 4 protocol ip 2 udp 53
```

Set CQ list 3: Assign packets transmitted from ASNC port 1 to queue 7.

```
Ruijie(config)# queue-list 3 interface serial 1 7
```

Set CQ list 9: Set the byte count of queue 10 to **1800**.

```
Ruijie(config)# queue-list 9 queue 10 byte-count 1800
```

Set CQ list 5: Set the queue length of queue 10 to **40**.

```
Ruijie(config)# queue-list 5 queue 10 limit 40
```

Set the queueing policy of SYNC port 0 to CQ list 5.

```
Ruijie(config)# interface serial 0
Ruijie(config-if)# custom-queue-list 1
```

Configuring PQ

PQ Configuration Tasks

Configuring PQ

You can configure a maximum of 16 groups for PQ. That is, the List-number range is 1–16. In each group, you can specify queues of packets, queue length, and byte count that can be continuously sent after a queue obtains the controlling right of sending packets.

Determining the Maximum Capacity of PQ

There are four queues (each assigned with the high, medium, normal, and low priority) in each group of queue lists. To specify the maximum number of packets allowed in each of the priority queues, run the following command in global configuration mode:

Command	**Function**
Ruijie(config)# **priority-list** *list-number* **queue-limit** [high-limit [medium-limit [normal-limit [low-limit]]]]	Specify the maximum packet count in a PQ queue.

List-number: indicates the number of a queue list within 1–16.

The following table describes the default queue lengths.

Queue	**Length**
high	20
medium	40
normal	60
Low	80

Note

To configure the interface congestion management policy, ensure that the fast-switching function configuration (enable or disable the function) is consistent on all system interfaces. Otherwise, the congestion management policy becomes invalid.

Assigning Packets to PQ Queues

You can specify multiple assignment rules. The **priority-list** commands are read in order of appearance until a matching protocol or interface type is found. When a match is found, the packet is assigned to the appropriate queue and the search ends. Packets that do not match other assignment rules are assigned to the default queue. To specify which queue to place a packet in, run the following commands in global configuration mode:

Command	**Function**
Ruijie(config)# **priority-list** *list-number* **protocol** *protocol-name* {**high** | **medium** | **normal** | **low**} [queue-keyword] [keyword-value]	Assign a packet to a PQ queue based on the protocol type.
Ruijie(config)# **priority-list** list-number **interface** *interface-type interface-number* {high | medium | normal | low}	Assign a packet to a PQ queue based on the type of the interface where the packet enters the device.
Ruijie(config)# **priority-list** *list-number* **default** {**high** | **medium** | **normal** | **low**}	Assign a default PQ queue with the normal priority for those packets that do not match any other rule in the priority list.

List number indicates the number of a priority queue. **Protocol-name** indicates the name of a protocol, such as IP, PAD, and so on. The values and meanings of **Queue-keyword** and **Eyword-value** are the same as these of CQ.

Applying the PQ List on an Interface

To apply a PQ list on an interface, run the following command in the interface configuration mode:

Command	**Function**
Ruijie(config-if)# **priority-group** *list-number*	Set the queueing policy of an interface to a specified PQ list.

Note

Each interface can use only one queueing policy and one matching list.

Monitoring PQ

To view information about input and output queues when PQ is enabled on an interface, run the following commands in the privileged user mode:

Command	Function
Ruijie# **show queue pq**	Show configuration information about PQ queues.
Ruijie# **show queue interface** interface-name *interface-number*	Show interface statistic information about PQ queues.
Ruijie# **debug qos cq**	Enable PQ debugging.

Note

For routers of the RSR series, you can view statistic information about fast-switching PQ queue interfaces by running the **show queue interface** command. "Qos Ref queue information" identifies statistic information about fast-switching PQ queue interfaces.

PQ Configuration Examples

Set PQ list 3: Assign packets transmitted from ASNC port 1/1 to the medium priority queue.

```
Ruijie(config)# priority-list 3 interface serial 1/1 medium
```

Set PQ list 6: Assign packets with a byte count greater than 250 to the medium priority queue.

```
Ruijie(config)# priority-list 6 protocol ip medium gt 250
```

Set PQ list 11: Assign IP packets with a byte count smaller than 250 to the medium priority queue.

```
Ruijie(config)# priority-list 11 protocol ip medium lt 250
```

Set PQ list 7: Assign packets matching IP access list 101 to the high priority queue.

```
Ruijie(config)# priority-list 7 protocol ip high list 101
```

Set PQ list 6: Assign Telnet packets to the high priority queue.

```
Ruijie(config)# priority-list 6 protocol ip high tcp 23
```

Set PQ list 6: Assign UDP domain name service packets to the low priority queue.

```
Ruijie(config)# priority-list 6 protocol ip low udp 53
```

Set PQ list 1: Assign packets that do not match any other rules in the priority list to the medium priority queue.

```
Ruijie(config)# priority-list 1 default medium
```

Set PQ list 2: set the lengths of the high, medium, normal, and low priority queues to 10, 40, 60, and 80.

```
Ruijie(config)# priority-list 2 queue-limit 10 40 60 80
```

Configuring Hold-queue

Configuring hold-queue

Run this command to set queue length on an interface in interface configuration mode.

Command	Function
Ruijie(config-if)# **hold-queue** *queue length* { **in** | **out** }	Sets queue length on an interface.

Queue length refers to the threshold of packets in a queue. When the number of packets reaches the threshold, the coming packets will be discarded.

The default values for input queue and output queue are 75 and 40 respectively.

Caution This command is used to modify three color-based thresholds of a queue for interface congestion and prevent green packets from being discarded preferentially. The default settings are applied generally. When cached packets exceed the red threshold, you should modify the value to make the number of cached packets below the red threshold.

Monitoring hold-queue

Run this command to show input and output queues information.

Command	Function
Ruijie# **show queue interface** *interface-name interface-number* [*queue-number*]	Shows queue information on an interface.

hold-queque Configuration Example

Configure fair queueing on synchronous interface 0 and set the threshold of packet number in an input queue to 128. When the packet number reaches the threshold, coming packets will be discarded.

```
Ruijie(config)# interface Serial 0
Ruijie(config-if)# hold-queue 128 in
```

Traffic Policing and Shaping

What Are Traffic Policing and Traffic Shaping

Traffic policing is used to control the rate of classified traffic flowing across an interface.

Traffic shaping allows you to control the burst traffic going out an interface to ensure that packets are delivered at stable rates and the network traffic remains stable.

Note NPE80 does not support traffic policing and shaping.

Overview of Traffic Policing and Shaping

- About Traffic Policing

Ruijie series of devices use the committed access rate (CAR) technology to limit the rate of traffic accessing the network to the committed rate. The rate-limiting feature and the packet classification feature can be configured together.

CAR uses the token bucket algorithm. You can set the capacity of the token bucket. If packets satisfy the preconfigured match rules, the token bucket accepts and processes the packets. If packets do not satisfy the preconfigured match rules, the token bucket refuses the packets and they continue to be transmitted. If there are sufficient tokens, packets processed by the token bucket continue to be transmitted. If tokens are insufficient, packets are discarded.

Ruijie series of devices support the configuration of at least 1-KB flow limit, the binding of CAR and ACL, and enable you to perform rate-limiting and traffic classification at the same time.

Ruijie series of devices support the service flow-limiting function based on IP applications. This enables you to limit service traffic flexibly based on IP applications' requirements on service traffic. For example, you can implement the service flow-limiting function based on the IPSec VPN tunnel.

- About Traffic Shaping

Ruijie series of devices shape traffic that is irregular or matches no predefined traffic features through the GTS (Generic Traffic Shaping) to facilitate bandwidth assignment between the upstream and downstream traffic on the network. GTS shapes traffic by buffering high-speed outbound traffic flow in the buffering area by constraining traffic to a particular bit rate using the token bucket.

Traffic shaping of routers is performed based on the following data flows:

1. All data flows passing physical interfaces
2. Data flows that are classified by using the standard or extended ACLs

- About the Token Bucket

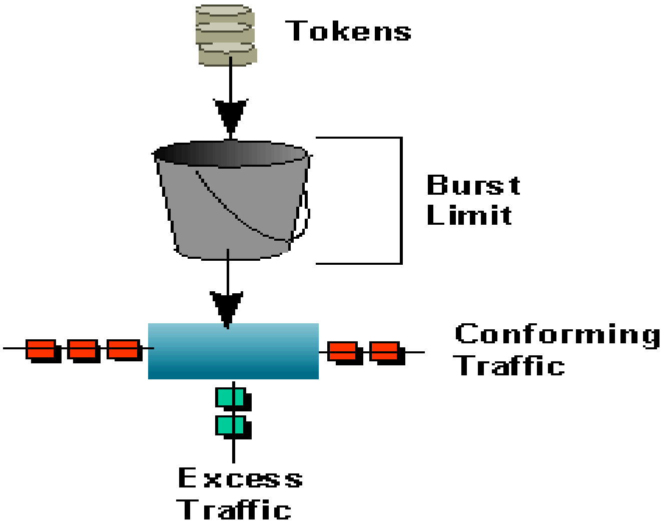
The token bucket algorithm is based on an analogy of a buffer (bucket) into which tokens, normally representing virtual data, are added at a fixed rate.

After entering the bucket, each token collects a packet from data queues and then is deleted from the bucket. This algorithm is associated with the token flow and the data flow. The following are three scenarios:

1. The data flow arrives at TBF using a rate equal to the rate of the token flow. In this case, each packet obtains a token and passes the queue without delay.
2. The data flow arrives at TBF using a rate smaller than the rate of the token flow. In this case, packets use some of the tokens and remaining tokens accumulate in the bucket until the bucket is full. Remaining tokens need to be consumed by the data flow arriving at TF using a rate greater than the rate of the token flow. In this case, burst transmission occurs.
3. The data flow arrives at TBF using a rate greater than the rate of the token flow. In this case, the bucket runs out of tokens quickly. This causes TBF interrupted for a period, namely, "threshold crossing". If packets arrive continuously, packets loss occurs.

The following figure shows the process.

Figure 7



What exactly is a token?

A token is the rate calibrated scale related to the value of CIR. Suppose that you configure an interface with a CIR rate of 8000 bit/s. Then, this interface can send 1000-bytes packets per second. At this time, the token equals to 1000 bytes, indicating that a token equal to 1000 is updated per second. Then, you can use the token to monitor whether packets accessing the interface exceed the committed rate.

Note

Ensure that the capacity of the token bucket is configured based on traffic bursts on the network. You need to expand the capacity of the token bucket to enhance the QoS tolerance capability for burst traffic on the network where burst traffic such as video or file transfer exists. Normally, the token bucket should at least support the 200-ms buffer capacity, that is, (CIR/8)*200ms.

Note

When GTS is used in rate-limiting, the frame gap and CRC are taken into account. The GRS rate limiting calculation is as follows:

1) Packet Per Second (PPS) = GTS rating limiting value in bps / ((packet length + frame gap + CRC) x 8); Round down the result for accuracy.

2) Receiver rate: PPS x actually received packet count in bytes x 8

Configuration Tasks of Traffic Policing

To configure the CAR traffic policing feature, run the following commands in the interface configuration mode:

Command	Function
Ruijie(config)# **interface** *interface-type interface-number*	Specify an interface where CAR rate limiting is to be enabled.
Ruijie(config-if)# **rate-limit** {**input** | **output**} **bps** burst-normal burst-max **conform-action** action exceed-action *action*	Perform packet rate-limiting for input or output packets of all traffic accessing the interface.

Input|output: indicates the input or output traffic that you need to limit.

Bps: indicates the maximum threshold rate (in bps) that you need to set for traffic.

Burst-normal burst-max: indicates the capacity of the token bucket in bytes.

Conform-action: indicates the processing policy for traffic that conforms to the rate limit.

Exceed-action: indicates the processing policy for traffic that exceeds the rate limit.

Action: indicates a processing policy. The processing policies are described as follows:

- Proceed to match the next policy.
- **Continue**: Match the next policy.
- **Drop**: Discard the packet.
- **Set-dscp-continue**: Continue to match the packet with the next policy after the DSCP domain is set for the packet.
- **Set-dscp-transmit**: Transmit the packet after the DSCP domain is set for the packet.
- **Set-prec-continue**: Continue to match the packet with the next policy after the IP Precedence domain is set for the packet.
- **Set-prec-transmit**: Transmit the packet after the IP Precedence domain is set for the packet.
- **Transmit**: Transmit the packet.

Caution

After IPSec encryption is enabled, the inbound traffic monitoring CAR will not support the packet revising policies including **set-dscp-continue**, **set-prec-continue**, **set-dscp-transmit**, and **set-prec-transmit,**

To configure the CAR rate limit feature for different types of traffic based on ACL or DSCP values, run the following commands in the interface configuration mode:

Command	Function
Ruijie(config)# **access-list** *acl-index*	Configure the ACL matching traffic.
Ruijie(config)# **interface** *interface-type* interface-number	Specify an interface where CAR rate limiting is to be enabled.
Ruijie(config-if)# **rate-limit** {**input** | **output**} [**access-group** *acl-index*] *bps* burst-normal burst-max **conform-action** action exceed-action *action*	Perform packet rate-limiting for the input or output packets of the traffic that matches the Acl-index access list.
Ruijie(config-if)# **rate-limit {input | output}** [**dscp** dscp-value] bps *burst-normal burst-max* **conform-action** *action* exceed-action *action*	Perform packet rate-limiting for the input or output packets of the traffic that matches the DSCP code value.

Caution

If multiple ACL CARs are configured and each flow matches an ACL, all matched flows take effect. If a flow is configure with ACL1 and ACL2, ACL1 takes effect. If a flow is configured with the same ACL rules and different action, all ACL rules take effect.

To configure the service rate limiting based on IP applications, run the following commands in the global configuration mode:

Command	Function
Ruijie(config)# **flow-limit** {**input** | **output**} **label** label-num *bps* burst-normal burst-max **conform-action** action **exceed-action** *action*	Configure the global service rate limiting template.
Ruijie(config)# **crypto map** *map-name seq-num* ipsec-isakmp	Enter the IPSec policy-map mode.
Ruijie(config-crypto-map)# **flow-label** *label-num*	Specify the flow label of the IPSec policy-map.

Input|output: indicates the input or output traffic that you need to limit.

label-num: indicates the label number the service rate limiting feature needs to match.

Bps: indicates the maximum threshold rate (in bps) that you need to set for traffic.

Burst-normal burst-max: indicates the capacity of the token bucket in bytes.

Conform-action: indicates the processing policy for traffic that conforms to the rate limit.

Exceed-action: indicates the processing policy for traffic that exceeds the rate limit.

Action: indicates a processing policy. The processing policies are described as follows:

Drop: Drop the packet.

Transmit: Transmit the packet.

Configuration Tasks of Traffic Shaping

To configure the GTS feature, run the following commands in the interface configuration mode:

Command	Function
Ruijie(config)# **interface** *interface-type interface-number*	Specify an interface where traffic shaping is to be enabled.
Ruijie(config-if)# **traffic-shape rate** *bit-rate* [*burst-size*] [*excess-burst-size*] [*buffer-limit*]	Shape the entire traffic of an interface.

Bit-rate: indicates the maximum threshold rate (in bps) that you need to shape. The maximum value is **1000000000**, indicating 1 Gbps.

Burst-size: indicates the size of burst packets that can be transmitted during each interval. The unit is bit.

Excess-burst-size: indicates the size of burst packets that can be exceedingly transmitted during first interval. The unit is bit.

Buffer-limit: indicates the buffer size of a GTS buffer queue. The default value is **1000**.

Note

The system processing traffic shaping policy takes effect based on an interface. After GTS is enabled on the interface, GTS must be enabled on all subinterfaces of the interface. Otherwise, the subinterfaces transmit data unevenly.

Note

After traffic shaping is enabled on an interface, **Burst-size** must be an integral multiple of a certain value (the packet size sent in 10 ms by using the traffic shaping rate). Otherwise, the system invokes the packet configuration parameter to round off **Burst-size** to the certain value to make the parameter take effect legally.

Note

Buffer queue configuration depends on network requirements. When forwarded data is delay-sensitive, you can decrease the queue depth to lower delay. When the forwarded data is burst or contains many small packets, you can increase the queue depth to improve the system buffer capability.

Note

When RSR30-44 performing large MSTP nodes aggregation, GTS rate might be slightly inaccurate for the aggregation capacity of RSR30-44 is limited.

To configure the GTS feature for different types of traffic based on ACLs, run the following commands in the interface configuration mode:

Command	Function
Ruijie(config)# **access-list** *acl-index*	Create the ACL matching traffic.
Ruijie(config)# **interface** *interface-type* interface-number	Specify an interface where GTS is to be enabled.
Ruijie(config-if)# **traffic-shape group** access-list bit-rate [burst-size [excess-burst-size]]	Perform interface traffic shaping for the input or output packets of the traffic that matches the acl-index access list.

Note

The **traffic-shape group** command and the **traffic-shape rate** command are mutually exclusive on an interface**.** That is, you cannot configure the two commands on the same interface.

Note

Ensure that the fast-switching function is disabled on the interface that deploys the traffic shaping function associated with ACL classification. This is required by in the current software version.

Configuration Tasks of Traffic Policing on a Policy-Map

To configure CAR rate limiting of a single rate on a policy-map, run the following commands:

Command	Function
Ruijie(config)# **policy-map** *policy-map-name*	Access and create a policy-map.
Ruijie(config-pmap)# **class** *class-map-name*	Use class-maps that have been defined.
Ruijie(config-pmap-c)# **police cir** [*bps*] [*burst-normal*] [*burst-max*] **conform-action** [*action*] **exceed-action** [*action*] **violate-action** [*action*]	Deploy the token bucket rate limiting of a single rate for this type of traffic.
Ruijie(config-if)# service-policy output policy-map-name	Specify the policy-map to be applied.

CIR: indicates the maximum threshold rate (in bps) that you need to set for traffic.

Burst-normal burst-max: indicates the capacity of the token bucket in bytes.

Conform-action: indicates the processing policy for traffic that conforms to the rate limit.

Exceed-action: indicates the processing policy for traffic that exceeds the rate limit.

violate-action: indicates the processing policy for traffic that exceeds the rate limit set for the second token bucket when there are two token buckets.

Action: indicates a processing policy. The processing policies are described as follows:

- **Drop**: Drop the packet.

- **Set-dscp-transmit**: Transmit the packet after the DSCP domain is set for the packet.
- **Set-prec-transmit**: Transmit the packet after the IP Precedence domain is set for the packet.
- **Transmit**: Transmit the packet.

To configure CAR rate limiting of two rates on a policy-map, run the following commands:

Command	**Function**
Ruijie(config)# **policy-map** *policy-map-name*	Access and create a policy-map.
Ruijie(config-pmap)# **class** *class-map-name*	Use class-maps that have been defined.
Ruijie(config-pmap-c)# **police cir** [bps] **pir** [*bps*] [*burst-normal*] [*burst-max*] **conform-action** [*action*] **exceed-action** [*action*] **violate-action** [*action*]	Deploy the token bucket rate limiting of two rates for this type of traffic.
Ruijie(config-if)# **service-policy output** policy-map-name	Specify the policy-map to be applied on the interface.

CIR: indicates the maximum threshold rate (in bps) that you need to set for traffic.

PIR: indicates the peak maximum threshold rate (in bps) that you need to set for traffic.

Burst-normal burst-max: indicates the capacity of the token bucket in bytes.

Conform-action: indicates the processing policy for traffic that conforms to the rate limit.

Exceed-action: indicates the processing policy for traffic that exceeds the rate limit.

violate-action: indicates the processing policy for traffic that exceeds the rate limit set for the second token bucket when there are two token buckets.

Action: indicates a processing policy. The processing policies are described as follows:

Drop: Drop the packet.

Set-dscp-transmit: Transmit the packet after the DSCP domain is set for the packet.

Set-prec-transmit: Transmit the packet after the IP Precedence domain is set for the packet.

Transmit: Transmit the packet.

Note

You can choose to use one of the four token bucket algorithms for rate limit on a policy-map:
1. Single token bucket algorithm: Use this algorithm if you have not configured **violate-action** and the value of **burst-normal** is equal to the value of **burst-max**.
2. Load mode of the single token bucket algorithm: Use this mode if you have not configured **violate-action** and the value of **burst-normal** is smaller than the value of **burst-max**.
3. Single-rate two-token-buckets algorithm: Use this algorithm if you have configured **violate-action** but not **pir**.
4. Two-rate two-token-bucket algorithm: Use this algorithm if you have configured both **violate-action** and **pir**.

Configuration Tasks of Traffic Shaping on a Policy-Map

To configure traffic shaping with an average rate on a policy-map, run the following commands:

Command	**Function**
Ruijie(config)# **policy-map** *policy-map-name*	Access and create a policy-map.
Ruijie(config-pmap)# **class** *class-map-name*	Use class-maps that have been defined.
Ruijie(config-pmap-c)# **shape average cir** [*bps*] [*bc*] [*be*]	Deploy traffic shaping with an average rate for this type of traffic.
Ruijie(config-if)# service-policy output policy-map-name	Specify the policy-map to be applied on the interface.

Bit-rate: indicates the maximum threshold rate (in bps) that you need to set for traffic shaping.

BC: indicates the size of burst packets that can be transmitted during each interval. The unit is bit.

BE: indicates the size of burst packets that can be exceedingly transmitted during first interval. The unit is bit.

To configure traffic shaping with a peak rate on a policy-map, run the following commands:

Command	Function
Ruijie(config)# **policy-map** *policy-map-name*	Access and create a policy-map.
Ruijie(config-pmap)# **class** *class-map-name*	Use class-maps that have been defined.
Ruijie(config-pmap-c)# **shape peak** [*bps*] [*bc*] [*be*]	Deploy traffic shaping with a peak rate for this type of traffic.
Ruijie(config-if)# **service-policy output** *policy-map-name*	Specify the policy-map to be applied on the interface.

Bit-rate: indicates the maximum threshold rate (in bps) that you need to set for traffic shaping.

BC: indicates the size of burst packets that can be transmitted during each interval. The unit is bit.

BE: indicates the size of burst packets that can be exceedingly transmitted during first interval. The unit is bit.

To configure the buffer size of traffic shaping on a policy-map, run the following commands:

Command	Function
Ruijie(config)# **policy-map** *policy-map-name*	Access and create a policy-map.
Ruijie(config-pmap)# **class** *class-map-name*	Use class-maps that have been defined.
Ruijie(config-pmap-c)# **shape max-buffers** [number-of-buffers]	Configure the buffer size of traffic shaping.
Ruijie(config-if)# **service-policy output** policy-map-name	Specify the policy-map to be applied on the interface.

Number-of-buffers: indicates the buffer size of traffic shaping. The default value is **1000**.

Note

Configurations of **shape average bps** and **shape peak bps** bring different traffic shaping results.
In traffic shaping, the peak rate is greater than the average rate. The calculation is as follows:
Peak rate = cir (1+ bc/be)

Note

Ensure that the fast-switching function is disabled on the interface that deploys the traffic shaping function associated with a policy map. This is required by in the current software version.

Configuring Car Token Bucket Algorithm Mode

There are two modes for calculating Car rate-limit bandwidth: 1) frame-gap-included mode; 2) frame-gap-excluded mode. By default, frame gap is excluded.

To configure car token bucket algorithm mode, perform the following steps:

Command	Function
Ruijie(config)# **bandwidth-mode car** **link-header | frame-gap**	Set Car rate-limit bandwidth calculation mode: link-header means frame gap is excluded when bandwidth is calculated; frame-gap means frame gap is included when bandwidth is calculated; By default, frame gap is excluded under bandwidth calculating mode.

Note

Car token bucket algorithm mode takes effect to both police car and rate-limit in the meantime.

Configuration Examples of Traffic Policing

Configuration Examples of Traffic Policing for Entire Traffic of an Interface

In the following examples, CAR traffic policing is configured on the ingress interface and the egress interface:

# Configure CAR traffic policing of egress interface packets on a serial interface.

# Limit the egress interface traffic to 300 kbps. If the traffic conforms to the rate limit, the traffic is transmitted. If the traffic exceeds the rate limit, the traffic is discarded.

```
interface Serial1/0
ip address 192.168.20.3 255.255.255.0
encapsulation ppp
rate-limit output 300000 3000 3000 conform-action  transmit exceed-action  drop
```

# Configure CAR traffic policing of ingress interface packets on a FastEthernet interface.

# Limit the ingress interface traffic to 2 Mbps. If the traffic conforms to the rate limit, the traffic is transmitted. If the traffic exceeds the rate limit, the traffic is discarded.

```
interface FastEthernet 0/0
ip address 192.168.20.3 255.255.255.0
encapsulation ppp
rate-limit input 2000000 3000 3000 conform-action  transmit exceed-action  drop
```

Configuration Examples of Traffic Policing for Traffic Satisfying Certain Conditions

In the following examples, CAR traffic policing is configured for egress traffic that satisfies certain conditions:

# Configure ACLs based on different TCP and UDP ports.

```
access-list 101 permit tcp any any eq 2065
access-list 102 permit udp any any range 16384 32767
access-list 103 permit tcp any any eq 1352
access-list 104 permit tcp any any eq www
access-list 105 permit tcp any any eq ftp-data
```

# Configure CAR traffic policing of egress interface packets based on ACLs on a serial interface.

```
interface Serial1/0
ip address 192.168.20.3 255.255.255.0
encapsulation ppp
rate-limit output access-group 101 256000 5000 5000 conform-action  transmit exceed-action
set-dscp-transmit 46
rate-limit output access-group 102 200000 3000 3000 conform-action  transmit exceed-action
set-prec-transmit 5
rate-limit output access-group 103 128000 3000 3000 conform-action  transmit exceed-action
set-prec-transmit 1
rate-limit output access-group 104 64000 3000 3000 conform-action  transmit exceed-action
drop
rate-limit output access-group 105 32000 3000 3000 conform-action  transmit exceed-action
drop
```

# Configure CAR traffic policing of egress interface packets based on DSCP code on a serial interface.

# Limit rates of traffic conforming to ACLs to 256 kbps, 200 kbps, 128 kbps, 64 kbps, and 32 kbps.

```
interface Serial1/0
ip address 192.168.20.3 255.255.255.0
encapsulation ppp
rate-limit output dscp 46 256000 5000 5000 conform-action  transmit exceed-action
set-dscp-transmit 46
rate-limit output dscp 10 200000 3000 3000 conform-action  transmit exceed-action
set-prec-transmit 5
rate-limit output dscp 18 128000 3000 3000 conform-action  transmit exceed-action
set-prec-transmit 1
rate-limit output dscp 20 64000 3000 3000 conform-action  transmit exceed-action  drop
```

Note

Do not set **token bucket** to a value that is too small. Otherwise, the system automatically adjust **token bucket** to a default value.

If you do not need to use the exceeding token bucket algorithm, you need to set the value of **Burst-normal** to be equal to or greater than the value of **burst-max**. If is normally feasible when the value of **Burst-normal** is equal to the value of **burst-max**.

Configuration Tasks of Traffic Policing on a Policy-Map

In the following examples, traffic policing based on policy maps is enabled for egress interface traffic satisfying conditions, and rate limit is enabled for each type of traffic using the single-rate two-token-buckets algorithm.

```
access-list 101 permit udp any any eq 100
access-list 102 permit udp any any eq 200
access-list 103 permit udp any any eq 300
access-list 104 permit udp any any eq 400
!
class-map match-all a1
match access-group 101
class-map match-all a2
match access-group 102
class-map match-all a3
match access-group 103
class-map match-all a4
match access-group 104
!
policy-map police
class a1
police cir 80000 2000 2000 conform-action transmit exceed-action drop violate-action drop
class a2
police cir 160000 2000 2000 conform-action transmit exceed-action drop violate-action drop
class a3
police cir 320000 6000 6000 conform-action transmit exceed-action drop violate-action drop
class a4
police cir 640000 6000 6000 conform-action transmit exceed-action drop violate-action drop
!
interface Serial1/0
ip address 192.168.20.3 255.255.255.0
encapsulation ppp
service-policy output police
```

In the following examples, traffic policing based on policy maps is enabled for egress interface traffic satisfying conditions, and rate limit is enabled for each type of traffic using the two-rate two-token-bucket algorithm.

```
!
policy-map police
class a1
police cir 80000 pir 100000 2000 2000 conform-action transmit exceed-action drop violate-action
drop
class a2
police cir 160000 pir 200000 2000 2000 conform-action transmit exceed-action drop
violate-action drop
class a3
police cir 320000 pir 400000 6000 6000 conform-action transmit exceed-action drop
violate-action drop
class a4
police cir 640000 pir 700000 6000 6000 conform-action transmit exceed-action drop
violate-action drop
!
interface Serial1/0
ip address 192.168.20.3 255.255.255.0
encapsulation ppp
service-policy output police
```

Configuration Examples of Traffic Shaping

Configuration Examples of Traffic Shaping for Entire Traffic of an Interface

# Configure GTS traffic shaping of egress interface packets on a serial interface.

# Shape the egress interface traffic to 300 kbps. If the traffic conforms to the rate limit, the traffic is transmitted. If the traffic exceeds the rate limit, the traffic is discarded. In this way, traffic is shaped.

```
interface Serial1/0
ip address 192.168.20.3 255.255.255.0
encapsulation ppp
traffic-shape rate 300000 9000 9000 1000
```

Configuration Examples of Traffic Shaping for Traffic Satisfying Certain Conditions

In the following examples, GTS is configured for egress traffic that satisfies certain conditions:

# Configure ACLs based on different TCP and UDP ports.

```
access-list 101 permit tcp any any eq 2065
access-list 102 permit udp any any range 16384 32767
access-list 103 permit tcp any any eq 1352
access-list 104 permit tcp any any eq www
access-list 105 permit tcp any any eq ftp-data
```

# Configure GTS of egress interface packets based on ACLs on a serial interface.

```
# Shape rates of traffic conforming to ACLs to 256 kbps, 200 kbps, 128 kbps, 64 kbps, and 32
kbps.
interface Serial1/0
ip address 192.168.20.3 255.255.255.0
encapsulation ppp
traffic-shape group 101 256000 10240 10240 1000
traffic-shape group 102 200000 8000 8000 1000
traffic-shape group 103 128000 10240 10240 1000
traffic-shape group 104 64000 12800 12800 1000
traffic-shape group 105 32000 12800 12800 1000
```

Note

In traffic policing, tokens are updated in the token bucket every time packets enter the bucket.
In traffic shaping, tokens are updated in the token bucket at each fixed interval.
In traffic policing, you can configure processing polices for input and output packets at the ingress interface and egress interface.
In traffic shaping, buffering and rate limiting can be configured only for egress interface packets.

Configuration Tasks of Traffic Shaping on a Policy-Map

In the following examples, traffic shaping based on a policy map is enabled for egress interface traffic satisfying conditions, and traffic shaping is enabled for each type of traffic using the common single-token-bucket algorithm.

```
access-list 101 permit udp any any eq 100
!
class-map match-all a1
match access-group 101
!
policy-map shape
class a1
shape  average 100000
shape  max-buffers 200  //buffer size: 200
!
interface Serial1/0
ip address 192.168.20.3 255.255.255.0
encapsulation ppp
service-policy output shape
!
```

In the following examples, traffic shaping with a peak rate based on a policy map is enabled for egress interface traffic satisfying conditions, and traffic shaping with a peak rate is enabled for each type of traffic using the common single-token-bucket algorithm.

```
access-list 101 permit udp any any eq 100
!
class-map match-all a1
match access-group 101
!
policy-map shape
class a1
shape  peak 100000
shape  max-buffers 200  //buffer size: 200
!
interface Serial1/0
ip address 192.168.20.3 255.255.255.0
encapsulation ppp
service-policy output shape
!
```

Configuration Examples of Car Token Bucket Algorithm Mode

Configure the car rate limit of out-going interface of GigabitEthernet to be 300kbps

```
Ruijie(config)# int gigabitEthernet 3/1/1
Ruijie(config-if-GigabitEthernet 3/1/1)#rate-limit output 300000 3000 3000 conform-action
transmit exceed-action drop
Ruijie(config)# bandwidth-mode car frame-gap
```

Maintenance and Debugging of Traffic Policing and Shaping

To monitor status of traffic policing and shaping, run the following commands in the privileged user mode:

Command	**Function**
Ruijie# show rate-limit interfaces [*interface-name*]	Show the rate limit of an interface.
Ruijie# **show traffic-shape** [*interface-name*]	Show the traffic shaping configuration policy of an interface.
Ruijie# **show traffic-shape statistics** [interface-name]	Show the packet statistic information about an interface in actual traffic shaping.
Ruijie# **show traffic-shape queue**	Show the queue information of the entire traffic shaping policy that is configured.

The following are examples of **show rate-limit**:

```
Ruijie#show rate-limit
serial 1/0
Output
matches access-group 101
params:  256000 bps, 3000 limit, 3000 extended limit
conformed 0 packets, 0 bytes; action: transmit 0
exceeded 0 packets, 0 bytes; action: drop 0
cbucket 0, cbs 3000; ebucket 0 ebs 0
```

The following are examples of **show traffic-shape interfaces**:

# **Interval** indicates the interval for updating the token bucket. It is set to 30 ms in this example.

# **Byte Limit** indicates the packet count allowed to transmit per second. It is set to 2250 bytes in this example.

```
Ruijie#show traffic-shape
Interface serial 1/0
Access   Target    Byte   Sustain    Excess    Interval    Increment   Adapt
VC     List     Rate    Limit   bits/int   bits/int   (ms)     (bytes)     Active
-    -      300000       2250      9000      9000      30        1125     -
Ruijie#
```

The following are examples of **show raffic-shape statistics**:

```
Ruijie#show traffic-shape statistics
Interface serial 1/0
Acc. Queue   Packets   Bytes   Packets   Bytes      Shaping
List  Depth                  Delayed   Delayed    Active
 -     0          0      0      0      0         no
Ruijie#
```

The following are examples of **show traffic-shape queue**:

```
Ruijie#show traffic-shape queue
Traffic queued in shaping queue on serial 1/0
Traffic shape group: null
Queuing strategy: weighted fair
Output queue: 0/1000/64/0 (size/max total/threshold/drops)
Output queue num: 0/0 (now/max)
Ruijie#
```

To debug packet compression, run the following commands in the privileged user mode:

Command	Function
Ruijie# **debug tc in**	Debug the traffic control for ingress interface packets.
Ruijie# **debug tc out**	Debug the traffic control for egress interface packets.

Note

For routers of the RSR series, you can view the token information about fast-switching traffic shaping interfaces by running the **show queue interface** command. "Qos Ref queue information" identifies the fast-switching statistic information.
You can run the **show rate-limit interface** command to view traffic policing statistic information about fast-switching interfaces on routers of the RSR series.

Congestion Avoidance

Overview

Congestion avoidance techniques monitor network traffic loads in an effort to anticipate and avoid congestion at common network bottlenecks. Congestion avoidance is achieved through packet dropping.

Overdue congestion brings great risks to network resources, and operations must be performed to eliminate congestion. Here, congestion avoidance indicates technologies used to monitor network traffic (such as queues and memory buffer) usage in an effort to avoid network overloading by initiatively dropping packets in the event of network congestion.

Traditional Packet-Drop Policy – Tail-Drop

Traditional packet-drop policy indicates tail-drop. Tail drop treats all traffic equally and does not differentiate between classes of service. Queues fill during periods of congestion. When the output queue is full and tail drop is in effect, packets are dropped until the congestion is eliminated and the queue is no longer full.

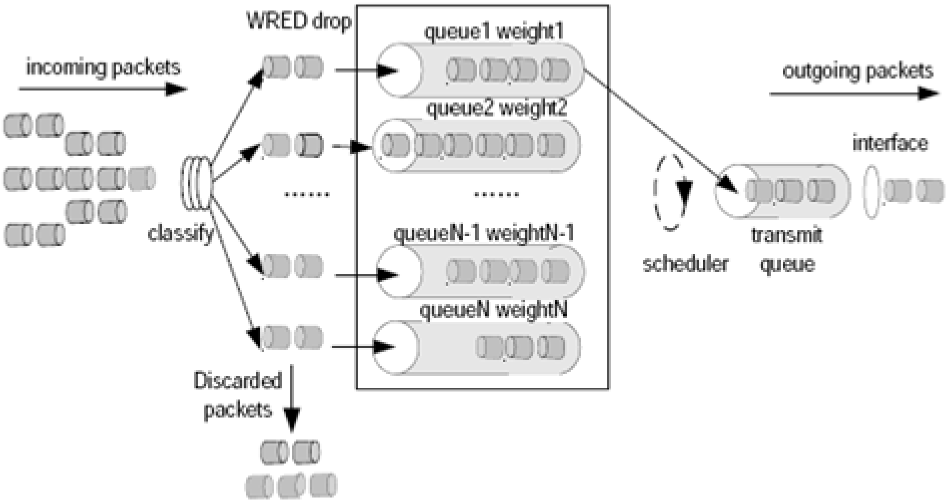
A host running TCP reduces the packet transmission rate when a lot of packets are lost and restore the packet transmission rate when the congestion is eliminated. This causes TCP global synchronization. When queues drop multiple TCP packets, hosts will reduce TCP traffic (referred to as slow start) in response, and then ramp up again simultaneously. When congestion is eased, traffic peaks occur. The two situations repeat one by one, causing cyclical periods of extreme congestion, followed by periods of under-utilization of the link.

WRED

WRED avoids TCP global synchronization by dropping packets randomly. When packets of a TCP connection are dropped and the transmission rate is reduced, other TCP connections still enjoy high transmission rates. In this case, there are TCP connections enjoy high transmission rates at any time, increasing the usage rate of line bandwidth.

WRED randomly drops queued packets after comparing the length, maximum threshold, and minimum threshold of a queue (configuring absolute length of the queue threshold). This is unfair for burst traffic and adverse to transmission. Therefore, the relative length of a queue is compared with the maximum threshold and minimum threshold (configuring the relative value for comparison between the queue threshold and the average length) for dropping packets. The average length of the queue is obtained by using the low-pass filter. It reflects the queue change trend but stays insensitive to the burst change of queue length, avoiding unfair treatment of burst traffic. The following figure shows the relationship between WRED and the queue mechanism.

Figure 8



Each queue is configured with a pair of maximum threshold and minimum threshold values. RED will drop packets using one of three methods:

- **No drop** – used when the queue length is smaller than the minimum threshold.
- **Drop all** – used when the queue length is greater than the maximum threshold.
- **Drop based on the WRED algorithm** – used when the queue length is between the maximum threshold and the minimum threshold.

Specifically: A random number is assigned to each packet that arrives. The random number is compared with the current mark probability denominator (MPD) of the queue. Packets are dropped if the random number is greater than the current MPD. The MPD increases in proportion to queue length, but there is a maximum MPD.

Configuration Tasks of Congestion Avoidance (WRED) Based on an Interface

To configure congestion avoidance based on precedence for an interface, run the following commands in the interface configuration mode:

Command	Function
Ruijie(config)# **interface** *interface-type interface-number*	Specify an interface where congestion avoidance is to be enabled.
Ruijie(config-if)# **random-detect prec-based**	Enable congestion avoidance based on precedence for the entire traffic of an interface.

To configure congestion avoidance based on DSCP classification, for an interface, run the following commands in the interface configuration mode:

Command	Function
Ruijie(config)# **interface** *interface-type interface-number*	Specify an interface where congestion avoidance is to be enabled.
Ruijie(config-if)# **random-detect** *dscp-based*	Enable congestion avoidance based on DSCP for the entire traffic of an interface.

Note

NPE80 does not support WRED congestion avoidance based on precedence and DSCP classification. NPE80 provides only the following command for enabling and disabling WRED:
Ruijie(config-if)# **random-detect**

Note To configure the interface congestion avoidance policy, ensure that the fast-switching function configuration (enable or disable the function) is consistent on all system interfaces. Otherwise, the congestion avoidance policy becomes invalid.

To configure the maximum threshold, the minimum threshold, and the MPD for each type of traffic based on DSCP classification, run the following commands in the interface configuration mode:

Command	Function
Ruijie(config)# **interface** *interface-type interface-number*	Specify an interface where congestion avoidance is to be enabled.
Ruijie(config-if)# **random-detect dscp** [*dscp-value*] [*min-threshold*] [*max-threshold*] [*mark-prob-denominator*]	Configure the maximum threshold, the minimum threshold, and the MPD for each type of traffic based on DSCP classification.

DSCP-value: indicates the value of DSCP. Traffic is classified based on this value.

Min-threshold: indicates the minimum drop threshold. The default values vary by traffic type.

Max-threshold: indicates the maximum drop threshold. The default values vary by traffic type.

Mark-prob-denominator: indicates the MPD that determines the number of packets that will be dropped. This is measured as a fraction, specifically 1/MPD. The greater the MPD is, the smaller the chance of each packet will be dropped. The default MPD is set to **10**, and one out of every 10 packets will be dropped. In other words, the chance of each packet being dropped is 10%.

To configure the maximum threshold, the minimum threshold, and the MPD for each type of traffic based on precedence classification, run the following commands in the interface configuration mode:

Command	Function
Ruijie(config)# **interface** *interface-type interface-number*	Specify an interface where congestion avoidance is to be enabled.
Ruijie(config-if)# **random-detect precedence** [*prec-value*] [*min-threshold*] [*max-threshold*] [*mark-prob-denominator*]	Configure the maximum threshold, the minimum threshold, and the drop probability for each type of traffic based on precedence classification.

Prec-value: indicates the value of Precedence. Traffic is classified based on this value.

Min-threshold: indicates the minimum drop threshold. The default values vary by traffic type.

Max-threshold: indicates the maximum drop threshold. The default values vary by traffic type.

Mark-prob-denominator: indicates the drop probability that determines the number of packets that will be dropped. This is measured as a fraction, specifically 1/MPD. The greater the MPD is, the smaller the chance of each packet will be dropped. The default MPD is set to **10**, and one out of every 10 packets will be dropped. In other words, the chance of each packet being dropped is 10%.

Note NPE80 does not support WRED congestion avoidance based on precedence and DSCP classification. You can run the following command to set the maximum threshold and the minimum threshold:
Ruijie(config-if)# **random-detect** [*min-threshold*] [*max-threshold*] [*mark-prob-denominator*]

To configure the weighting factor of congestion avoidance for an interface, run the following commands in the interface configuration mode:

Command	Function
Ruijie(config)# **interface** *interface-type interface-number*	Specify an interface where congestion avoidance is to be enabled.
Ruijie(config-if)# **random-detect exponential-weighting-constant** [*exponential-value*]	Configure the weighting factor of congestion avoidance for an interface.

Exponential-value: indicates the weighting factor. The default value is **9**. The greater the value is, the smaller the MPD is.

Note If the queueing algorithm of an interface is not FIFO, you need to cancel the current queueing algorithm before configuring WRED congestion avoidance for the interface.

Configuration Tasks of Congestion Avoidance (WRED) Based on a Policy-Map

Note NPE80 does not support congestion avoidance (WRED) based on a policy-map.

To configure congestion avoidance based on precedence for a policy-map, run the following commands:

Command	Function
Ruijie(config)# **policy-map** *policy-map-name*	Access and create a policy-map.
Ruijie(config-pmap)# **class** *class-map-name*	Use class-maps that have been defined.
Ruijie(config-if)# **random-detect prec-based**	Enable congestion avoidance based on precedence for the entire traffic of an interface.

To configure congestion avoidance based on DSCP classification for a policy-map, run the following commands:

Command	Function
Ruijie(config)# **policy-map** *policy-map-name*	Access and create a policy-map.
Ruijie(config-pmap)# **class** *class-map-name*	Use class-maps that have been defined.
Ruijie(config-if)# **random-detect dscp-based**	Enable congestion avoidance based on DSCP for the entire traffic of an interface.

To configure the maximum threshold, the minimum threshold, and the MPD for each type of traffic based on DSCP classification, run the following commands:

Command	Function
Ruijie(config)# **policy-map** *policy-map-name*	Access and create a policy-map.
Ruijie(config-pmap)# **class** *class-map-name*	Use class-maps that have been defined.
Ruijie(config-if)# **random-detect dscp** [*dscp-value*] [*min-threshold*] [*max-threshold*] [*mark-prob-denominator*]	Configure the maximum threshold, the minimum threshold, and the MPD for each type of traffic based on DSCP classification.

DSCP-value: indicates the value of DSCP. Traffic is classified based on this value.

Min-threshold: indicates the minimum drop threshold. The default values vary by traffic type.

Max-threshold: indicates the maximum drop threshold. The default values vary by traffic type.

Mark-prob-denominator: indicates the drop probability that determines the number of packets that will be dropped. This is measured as a fraction, specifically 1/MPD. The greater the MPD is, the smaller the chance of each packet will be dropped. The default MPD is set to **10**, and one out of every 10 packets will be dropped. In other words, the chance of each packet being dropped is 10%.

To configure the maximum threshold, the minimum threshold, and the MPD for each type of traffic based on precedence classification, run the following commands:

Command	Function
Ruijie(config)# **policy-map** *policy-map-name*	Access and create a policy-map.
Ruijie(config-pmap)# **class** *class-map-name*	Use class-maps that have been defined.
Ruijie(config-if)# **random-detect precedence** [*prec-value*] [*min-threshold*] [*max-threshold*] [*mark-prob-denominator*]	Configure the maximum threshold, the minimum threshold, and the MPD for each type of traffic based on precedence classification.

Prec-value: indicates the value of Precedence. Traffic is classified based on this value.

Min-threshold: indicates the minimum drop threshold. The default values vary by traffic type.

Max-threshold: indicates the maximum drop threshold. The default values vary by traffic type.

Mark-prob-denominator: indicates the drop probability that determines the number of packets that will be dropped. This is measured as a fraction, specifically 1/MPD. The greater the MPD is, the smaller the chance of each packet will be dropped. The default MPD is set to **10**, and one out of every 10 packets will be dropped. In other words, the chance of each packet being dropped is 10%.

To configure the weighting factor of congestion avoidance for a policy-map, run the following commands:

Command	Function
Ruijie(config)# **policy-map** *policy-map-name*	Access and create a policy-map.
Ruijie(config-pmap)# **class** *class-map-name*	Use class-maps that have been defined.
Ruijie(config-if)# **random-detect exponential-weighting-constant** [*exponential-value*]	Configure the weighting factor of congestion avoidance for an interface.

Exponential-value: indicates the weighting factor. The default value is **9**. The greater the value is, the smaller the drop probability is.

Configuration Examples of Congestion Avoidance (WRED)

Configuration Examples of Congestion Avoidance for Entire Traffic of an Interface

In the following examples, WRED congestion avoidance is configured on a SYNC interface:

# Configure WRED congestion avoidance based on precedence classification for packets on a serial interface.

# Set that the minimum threshold to **5**, maximum threshold to **100**, and MPD to **10** for traffic with precedence 1.

# Set that the minimum threshold to **10**, maximum threshold to **100**, and MPD to **10** for traffic with precedence 2.

# Set that the minimum threshold to **20**, maximum threshold to **100**, and MPD to **10** for traffic with precedence 3.

# Set that the minimum threshold to **30**, maximum threshold to **100**, and MPD to **10** for traffic with precedence 4.

```
interface Serial1/0
ip address 192.168.20.3 255.255.255.0
encapsulation ppp
random-detect
random-detect precedence 1 5 100 10
random-detect precedence 2 10 100 10
random-detect precedence 3 20 100 10
random-detect precedence 4 30 100 10
```

Configuration Examples of Congestion Avoidance on a Policy-Map

In the following examples, WRED congestion avoidance is configured on a SYNC interface:

# Configure WRED congestion avoidance based on policy-map classification for packets on a serial interface.

# Set that the minimum threshold to **5**, maximum threshold to **100**, and MPD to **10** for traffic with DSCP af11.

# Set that the minimum threshold to **10**, maximum threshold to **100**, and MPD to **10** for traffic with DSCP af21.

# Set that the minimum threshold to **20**, maximum threshold to **100**, and MPD to **10** for traffic with DSCP af31.

# Set that the minimum threshold to **30**, maximum threshold to **100**, and MPD to **10** for traffic with DSCP af41.

```
access-list 101 permit udp any any eq 100
access-list 102 permit udp any any eq 200
access-list 103 permit udp any any eq 300
access-list 104 permit udp any any eq 400
class-map match-any b1
match access-group 101
match access-group 102
class-map match-any b2
match access-group 103
match access-group 104
policy-map random
class b1
```

```
bandwidth 900
random-detect dscp-base
random-detect dscp af11 5 100 10
random-detect dscp af21 10 100 10
class b2
bandwidth 900
random-detect dscp-base
random-detect dscp af31 20 100 10
random-detect dscp af41 30 100 10
interface Serial1/0
ip address 192.168.20.3 255.255.255.0
encapsulation ppp
service-policy output random
```

Maintenance of Congestion Avoidance

The following section describes maintenance of congestion avoidance based on an interface:

```
Ruijie# show queue interface s 1/2
Current random-detect configuration:
serial 1/2
Queuing strategy: random early detection (WRED)
Exp-weight-constant: 9 (1/512)
Mean queue depth: 81407
Avg arrive time: 3000
class          Random drop     Tail drop    Minimum Maximum  Mark
pkts/bytes      pkts/bytes    thresh  thresh  prob
0     0/0     0/0     20     40     1/10
1     336/81312     6174/1494108     5     100     1/10
2     288/69696     6168/1492656     10     100     1/10
3     238/57596     6175/1494350     20     100     1/10
4     112/27104     6321/1529682     30     100     1/10
5     0/0     0/0     31     40     1/10
6     0/0     0/0     33     40     1/10
7     0/0     0/0     35     40     1/10
Ruijie#
```

The following section describes maintenance of congestion avoidance based on a policy-map:

```
Ruijie# show policy-map interface s 1/2
serial 1/2  output : random1
Class b1
Current random-detect configuration:
Exp-weight-constant: 9 (1/512)
Mean queue depth: 65529
Avg arrive time: 3000
class        Random drop     Tail drop    Minimum Maximum  Mark
pkts/bytes      pkts/bytes    thresh  thresh  prob
0     0/0     0/0     20     40     1/10
1     739/178838     0/0     5     100     1/10
2     614/148588     0/0     10     100     1/10
3     0/0     0/0     26     40     1/10
4     0/0     0/0     28     40     1/10
5     0/0     0/0     31     40     1/10
6     0/0     0/0     33     40     1/10
7     0/0     0/0     35     40     1/10
Class b2
Current random-detect configuration:
Exp-weight-constant: 9 (1/512)
Mean queue depth: 65530
Avg arrive time: 3000
class     Random drop     Tail drop    Minimum Maximum  Mark
pkts/bytes      pkts/bytes    thresh  thresh  prob
0     0/0     0/0     20     40     1/10
1     0/0     0/0     22     40     1/10
2     0/0     0/0     24     40     1/10
3     394/95348     0/0     20     100     1/10
4     68/16456     0/0     30     100     1/10
5     0/0     0/0     31     40     1/10
6     0/0     0/0     33     40     1/10
7     0/0     0/0     35     40     1/10
```

```
serial 1/2  output : random1
Weighted Fair Queuing
Class b1
Output Queue: queue_num 265
Bandwidth 900 (kbps) Packets Matched 17188 Sended 5217 Max Thresh 64 (packets)
(discards/tail drops) 11971/489148 , weight 91
Class b2
Output Queue: queue_num 266
Bandwidth 900 (kbps) Packets Matched 17793 Sended 5218 Max Thresh 64 (packets)
(discards/tail drops) 12575/489148 , weight 91
Ruijie#
```

Compression Protocol

Packet Compression Protocols

When the TCP and UDP packets carry only a small-size payload but the total amount of IP/TCP or IP/UDP/RTP header information occupies a fixed 40 bytes, the bandwidth of the low-rate link is wasted. Therefore, a series of compression algorithms are needed to compress and decompress the IP/TCP, IP/UDP, or IP/UDP/RTP header information. This is why packet compression protocols are formulated.

Packet compression mainly includes Transmission Control Protocol (TCP) packet compression and User Datagram Protocol (UDP) packet compression. Ruijie products implement the compression in the following two packet formats:

- VJ TCP packet compression: It refers to the compression of TCP packets and is mainly applicable to the low-speed serial links. It complies with the standard RFC 1144 - Compressing TCP/IP Headers for Low-Speed Serial Links.
- IP Header Compression (IPHC): It is IP framework-based packet compression, which consists of three types of packet compression: IP/TCP, IP/UDP, and IP/UDP/RTP. It complies with the standard RFC 2057 - Source Directed Access Control on the Internet.

VJ TCP Packet Compression

VJ TCP packet compression refers to the compression of the TCP packets. It adopts the VJ TCP algorithm and complies with RFC 1144.

A TCP/IP data packet header occupies 40 bytes: 20 bytes for the IP header and 20 bytes for the TCP header. As the TCP and IP headers are not designed by the same organization and all of the fields in the headers are used for a specific purpose, it is impossible to simply ignore the fields for efficiency reasons.

However, TCP connections are created and each connection transmits dozens or even hundreds of packets. Then, how much of the information in each packet in one connection can remain the same? The answer is a half. The sender and receiver can track the connections, and the receiver of each connection can retain a copy of the header of the last received packet. When the sender sends a packet containing a small-size (no more than 8 bits) connection identifier, 20-byte changing content, and 20-byte unchanged content, the receiver can fill only the 20-byte unchanged content into the last saved header. In this way, the data packet is compressed into a half of the original size.

The TCP packet compression can release the bandwidth of the low-speed serial links, which is particularly effective for small packet services such as Telnet.

VJ TCP packet compression supports the following link-layer protocols:

- PPP: Point-to-Point Protocol (PPP): It needs to negotiate the compression protocol option during the IP Control Protocol (IPCP) packet negotiation.
- Serial Line Internet Protocol (SLIP)
- High-Level Data Link Control (HDLC)
- Frame Relay

IPHC

IPHC compresses all TCP and UDP packets in the IPHC format under the IP header compression framework.

At present, IPHC is mainly applied to voice data compression on a low-speed link, including the compression of the real-time transfer (RTP) data, UDP data, and TCP data.

In fact, IPHC can be used to compress the IP, UDP, and RTP headers to achieve a great success similar to that achieved by VJ TCP packet compression in TCP packets. The compression can be applied to the RTP header (in the end-to-end application), or the IP/UDP/RTP combined header (in the link-by-link application). Compressing the 40-byte combined header is more practically effective than compressing only the 12-byte RTP header, because the compressed packages in the two cases both occupy 2–4 bytes. In addition, due to the low packet delay and loss rate, the compression performance in the link-by-link application is better.

The IPHC solution is aimed to compress the IP/UDP/RTP header of most packets into 2 bytes when the UDP checksum is not sent, and into 4 bytes when the UDP checksum is sent.

The IPHC supports the following link-layer protocols:

- PPP: It needs to negotiate the compression protocol option during the IPCP packet negotiation.
- HDLC: Ruijie products support the IPHC compression for the TCP and RTP packets only, and do not support the TCP/RTP packet compression proposed by Cisco.

- Frame Relay: Ruijie products support the TCP/RTP packet compression with the local management interface (LMI) type set to Cisco.

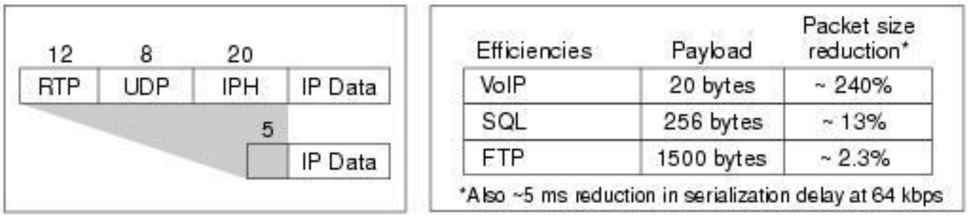
Efficiency and Inapplicable Scenarios of Packet Compression

The following analyzes the efficiency of the packet compression.

- For RTP packet compression, a 40-byte IP/UDP/RTP packet header is compressed to 2–4 bytes. It is assumed that the RTP payload is 24 bytes. In this case, the original 64-byte packet becomes about 27 bytes. The compression ratio is 2.3 times.
- For TCP packet compression, a 40-byte IP/TCP packet header is compressed to 2–4 bytes. It is assumed that the TCP payload is 24 bytes. In this case, the original 64-byte packet becomes about 27 bytes. The compression ratio is 2.3 times.

The following figure shows the efficiency comparison of RTP packet compression.

Figure 9



Efficiencies	Payload	Packet size reduction*
VoIP	20 bytes	~ 240%
SQL	256 bytes	~ 13%
FTP	1500 bytes	~ 2.3%

*Also ~5 ms reduction in serialization delay at 64 kbps

Packet compression protocols are inapplicable to Ruijie products in the following scenarios:

- The Ethernet interface does not support packet compression.
- Serial links with a rate higher than 2 Mbps do not support packet compression.
- Packet compression is not supported in the case with multi-link PPP.
- Fragmented IP packets cannot be compressed.
- Packet compression cannot be configured on the logical interface of the dialer, and must be configured on the physical interface.

Packet Compression Protocol Configuration

To configure TCP packet compression on an interface, run the following commands in interface configuration mode:

Command	Function
Ruijie(config-if)# **ip tcp header-compression**	Configures TCP/IP packet compression.
Ruijie(config-if)# **no ip tcp header-compression**	Cancels TCP/IP packet compression.

To configure the passive mode of TCP packet compression on an interface (the local packets are compressed only when the packets from the peer end are compressed), run the following commands in interface configuration mode:

Command	Function
Ruijie(config-if)# **ip tcp header-compression passive**	Configures the passive mode of TCP/IP packet compression.
Ruijie(config-if)# **no ip tcp header-compression**	Cancels TCP/IP packet compression.

To configure the number of connections for TCP packet compression on an interface, run the following commands in interface configuration mode:

Command	Function
Ruijie(config-if)# **ip tcp compression-connections** *number*	Configures the number of connections for TCP/IP packet compression.
Ruijie(config-if)# **no ip TCP compression-connections**	Cancels the connections for TCP/IP packet compression.

1.

Note The number of connections configured on the compression interfaces at both ends must be the same. Otherwise, the TCP/IP compression module may malfunction or even fail in communication.

To configure RTP packet compression on an interface, run the following commands in interface configuration mode:

Command	Function
Ruijie(config-if)# **ip rtp header-compression**	Configures IP/RTP packet compression.
Ruijie(config-if)# **no ip rtp header-compression**	Cancels IP/RTP packet compression.

To configure the passive mode of RTP packet compression on the interface (the local packets are compressed only when the packets from the peer end are compressed), run the following commands in interface configuration mode:

Command	Function
Ruijie(config-if)# **ip rtp header-compression passive**	Configures the passive mode of the IP/RTP packet compression.
Ruijie(config-if)# **no ip rtp header-compression**	Cancels IP/RTP packet compression.

To configure the number of connections for RTP packet compression on an interface, run the following commands in interface configuration mode:

Command	Function
Ruijie(config-if)# **ip rtp compression-connections** *number*	Configures the number of connections for IP/RTP packet compression.
Ruijie(config-if)# **no ip rtp compression-connections**	Cancels the connections for IP/RTP packet compression.

Note The number of connections configured the compression interfaces at both ends must be the same. Otherwise, the IP/RTP compression module may malfunction or even fail in communication.

Note The compression modes configured on the compression interfaces at both ends must be the same. For example, IP/RTP compression, or TCP/IP compression, or IP/RTP compression, or TCP/IP compression is configured at both ends. In addition, the compression formats of each compression mode must be the same at both ends, that is, both compression formats are IPHC format or IETF format. If the configurations on the compression interfaces at both ends are inconsistent, the communication may fail.

Packet Compression Configuration Examples

PPP-based Packet Compression

The following are examples of PPP-based packet compression:

#Configure the PPP-based RTP packet compression.

```
interface Serial1/0
ip address 192.168.20.3 255.255.255.0
encapsulation ppp
ip rtp header-compression IPHC-format
```

#Configure the PPP-based TCP packet compression.

```
interface Serial1/0
ip address 192.168.20.3 255.255.255.0
```

```
encapsulation ppp
ip tcp header-compression
```

HDLC-based Packet Compression

The following are examples of HDLC-based packet compression.

#Configure the HDLC-based RTP packet compression.

```
interface Serial1/0
ip address 192.168.20.3 255.255.255.0
ip rtp header-compression iphc-format
```

Note

Ruijie products only support HDLC-based packet compression in the IPHC format. If a Ruijie device interconnects with a Cisco device, packet compression in the IPHC format must also be configured on the Cisco device. If IP/RTP header compression is configured on the Cisco device, compression and decompression will fail, resulting in a communication failure.

Frame Relay-based Packet Compression

The following is an example of Frame Relay-based packet compression:

#Configure Frame Relay-based RTP and TCP packet compression.

```
interface Serial1/0
ip address 1.1.1.1 255.255.255.0
frame-relay map ip 1.1.1.2 20
frame-relay intf-type dce
frame-relay local-dlci 20
frame-relay lmi-type cisco
ip rtp header-compression
ip tcp header-compression
```

Note

Ruijie products and Cisco products both support the Frame Relay-based RTP packet compression with the LMI type set to Cisco.
Cisco products also allow users to enable or disable the compression function on different data links of different data link connection identifiers (DLCIs). Different from Cisco products, the policy of Ruijie products is to compress all, or none.

Packet Compression Maintenance and Debugging

To monitor packet compression, run the following commands in privileged EXEC mode:

Command	Function
Ruijie# **show ip rtp header-compression** *interface*	Displays related information about RTP packet compression and decompression.
Ruijie# **show ip tcp header-compression** *interface*	Displays related information about TCP packet compression and decompression.
Ruijie# **clear rtp-compress** [*interceface*]	Clears the counts of RTP packet compression and decompression.
Ruijie# **clear rtp-compress** [*interceface*]	Clears the counts of TCP packet compression and decompression.

The following is an example of output of the **show ip rtp header-compression** command:

```
RTP/UDP/IP header compression statistics:
Interface serial 1/0: active on
Rcvd:   407 total, 406 compressed, 0 errors
0 dropped, 406 buffer copies, 0 buffer failures
Sent:   406 total, 405 compressed,
14716 bytes saved, 8494 bytes sent
2.73 efficiency improvement factor
```

```
Connect: 256 rx slots, 256 tx slots, 0 long searches, 1 misses
99% hit ratio, five minute miss rate 0 misses/sec, 0 max
```

To debug packet compression, run the following commands in privileged EXEC mode:

Command	Function
Ruijie# **debug ip rtp header-compression**	Debugs the receiving and sending of compressed and decompressed RTP packets.
Ruijie# **debug ip rtp errors**	Debugs RTP packet compression errors.
Ruijie# **debug ip tcp header-compression**	Debugs the receiving and sending of compressed and decompressed TCP packets.

Configuring MPLS QOS

MPLS QOS

MPLS QoS provided by Ruijie Networks supports the following functions:

- Congestion management

Currently, MPLS experimental value based WFQ, CBWFQ, PQ and LLQ queues are supported.

- Traffic policing and traffic shaping

Currently, MPLS experimental value based car is supported to limit the bandwidth of data stream, specify actions for handling excess traffic, and limit traffic burst in respect of MPLS traffic shaping so that message flows can be transmitted at an even rate.

- Congestion avoidance

Currently, MPLS experimental value based RED and WERD are supported.

For more QoS related features, please refer to QoS configuration guideline

Congestion Management

Configuration of Weighted Fair Queuing (WFQ)

WFQ Configuration Tasks

When standard WFQ is used, the data packets will be classified through traffic. Ruijie routers currently support MPLS EXP value based traffic classification. WFQ allocates the same bandwidth for respective traffics. The traffic-based WFQ is also called fair queuing, as all traffic are provided with the same weight.

To configure WFQ, the following tasks need to be completed:

- Configuring WFQ
- Monitoring fair queuing

Note

Since the flash configurations varies from router to router, it is suggested to use "fair-queue" command to configure different queue depth and numbers.

Configuring WFQ

To configure WFQ, execute the following commands in interface configuration mode:

Command	**Function**
Ruijie(config-if)# **fair-queue** [*congestive-discard-threshold* [*dynamic-queues*]]	Configure WFQ
Ruijie(config-if)# no fair-queue	Remove WFQ configurations

WFQ parameters:

Parameter	**Description**
congestive-discard-threshold	The maximum number of packets (threshold) allowed in each queue (64 by default). When the number of packets reaches this threshold, the incoming new packets will be discarded. (This parameter is optional)
dynamic-queues	Number of dynamic queues. Default: 256; scope: integer between 16 and 4096, and must be the power of 2. (This parameter is optional)

Note

To configure WFQ congestion management policy on the interface, all interfaces of the system must enable fast forwarding function, otherwise this function will become invalid.

Note

We can find out the WFQ congestion management configurations of fast-forward IP QoS and MPLS QoS are same. The two functions may overlap on LSP Egress device, and the current software version will give preference to the WFQ function of MPLS QoS under such a circumstance.

Monitoring WFQ

To view WFQ configurations of interface, execute the following commands in privileged user mode.

Command	Function
Ruijie# **show queue interface** *interface–name interface-number [queue-number]*	Display the QoS queue information on the designated network interface.

WFQ Configuration Examples

As shown below, configure fair queuing on the synchronization interface: congestion drop threshold (threshold) being 128 packets and dynamic queues being 512.

```
interface Serial 3/0
 ip ref
 ip address 192.168.200.1 255.255.255.0
 mpls ip
 fair-queue 128 512
An example of viewing interface configurations in privileged user mode is shown below:
Ruijie# show queue interface serial 3/0

  Queueing strategy: weighted fair
  Output queue: 0/300/128/0 (size/max total/threshold/drops)
  Output queue num: 0/0/512 (now active/max active/max total)

  Qos Ref queue information
  Current Policy(s) : WFQ
  interface cir: 2048000
  Queueing strategy: weighted fair
    Dequeue threshold:  Green 25000, Yellow 37500, Red 50000
    Queues: Queues total len 0, MeanBurst 800
    Queues: gts gap 7, deta bits 262, token bucket 51200
    Queues: Max 19353 pkts, used 0 pkts
    Queues: rtpQ: 0 pkts, 0 bytes
    Queues: llQ:  0 pkts, 0 bytes
    Queues: genQ: 0 pkts, 0 bytes
    wfq para: cir(2048000), delta(13/2080)
    wfq_tb para: cir(3938), delta(7/129)
    Output queue: 0/0(send/drops)
```

From the above messages, it can be observed that the queuing policy of interface adopts WFQ and the congestion drop threshold (threshold) is 128.

Configuration of Class-based Weighted Fair Queueing (CBWFQ)

CBWFQ Configuration Tasks

In order to configure CBWFQ, the following tasks need to be completed:

Configuring CBWFQ

Defining Class Maps

This function is required in order to realize CBWFQ function. The user can define network packet classifying policy in the class map, and use these class maps by specifying the name of class map in the policy map. The same class map can be used by one or more policy maps. The typical configurations of this function are shown below:

Command	**Function**
Ruijie(config)# **class-map** *match-all class-map-name*	Enter/create AND-type class map (to meet all conditions in the class map).
Ruijie(config)# **class-map match-any** *class-map-name*	Enter/create OR-type class map (to meet only one condition in the class map).
Ruijie(config-cmap)# **match mpls experimental** *value* or Ruijie(config-cmap)# **match qos-group** *value* Ruijie(config-cmap)# **match not** *match-type value*	Configure network packet classifying policy (according to the group ID of message, EXP value of MPLS message or the false condition of the abovementioned classifying policies.
Ruijie(config-cmap)# **exit**	Exit class map configuration layer

Class-map-name: name of class map;

Match-all: To meet all conditions in the class map; the default type of class map is Match-all;

Match-any: to meet only one condition in the class map;

Mpls Experimental: the value of Experimental field of MPLS packets;

Qos-group: group ID of packets;

Not match-type: false condition of classifying policy.

Configuring Class Policy in the Policy Map

This function is required in order to realize CBWFQ function. In the policy map, the user can use all class maps configured on the device (up to 64 different class maps). The user may allocate bandwidth for the class map used, but the total bandwidth allocated for all used class maps must not exceed the bandwidth allocated to CBWFQ by the interface applied with this policy map. The typical configurations of this function are shown below:

Command	**Function**
Ruijie(config)# **policy-map** *policy-map-name*	Enter/create policy map
Ruijie(config-pmap)# **class** *class-map-name*	Use the class map defined.
Ruijie(config-pmap-c)# **bandwidth** { *bandwidth-kbps* | **percent** *percent-number* }	Allocate bandwidth for specific class of traffic
Ruijie(config-pmap-c)# **queue-limit** *number-of-packets*	Define queue depth
Ruijie(config-pmap-c)# **exit**	Exit class reference configuration layer
Ruijie(config-pmap)# **exit**	Exit policy map configuration layer

Policy-map-name: name of policy map;

Class-map-name: name of class map;

Bandwidth-kbps: bandwidth allocated (unit: kbps);

Percent-number: percentage of bandwidth allocated (in regard to all available bandwidth of network interface);

Number-of-packets: CBWFQ queue depth (maximum number of packets allowed).

Applying Service Policy on the Designated Interface

This function is required in order to realize CBWFQ function. Applying service policy on the designated interface will enable CBWFQ function, after which the class used by the corresponding policy map will have the corresponding queue. The typical configuration of this function is shown below:

Command	**Function**
Ruijie(config-if)# **service-policy output** *policy-map-name*	Enable CBWFQ and specify the policy map to be applied.

Policy-map-name: name of policy map.

Note	To configure policy map on the interface, all interfaces of the system must enable fast forwarding function, otherwise this function will become invalid.

Configuring Bandwidth for an Existing Class

This function is optional for CBWFQ. The typical configurations of this function are shown below:

Command	**Function**
Ruijie(config)# **policy-map** *policy-map-name*	Enter/create policy map
Ruijie(config-pmap)# **class** *class-map-name*	Use the class map defined.
Ruijie (config-pmap-c)# **bandwidth** { *bandwidth-kbps* | **percent** *percent*}	Allocate bandwidth for specific class of traffic

Policy-map-name: name of policy map;

Class-map-name: name of class map;

Bandwidth-kbps: bandwidth allocated (unit: kbps);

Percent-number: percentage of bandwidth allocated (in regard to all available bandwidth of network interface);

The user may allocate bandwidth for the specific type of network traffic. By default, 1% of bandwidth is allocated to the specific type of network traffic.

Configuring the Queue Depth for an Existing Class

This function is optional for CBWFQ. The typical configurations of this function are shown below:

Command	**Function**
Ruijie(config)# **policy-map** *policy-map-name*	Enter/create policy map
Ruijie(config-pmap)# **class** *class-map-name*	Use the class map defined.
Ruijie(config-pmap-c)# **queue-limit** *number-of-packets*	Define queue depth

Policy-map-name: name of policy map;

Class-map-name: name of class map;

Number-of-packets: CBWFQ queue depth (maximum number of packets allowed).

The user may configure queue depth for the corresponding CBWFQ queue of specific network traffic. The default value is 64, which means that after the corresponding CBWFQ queue has 64 packets, the system will discard subsequent network packets entering into this queue. By this time, Ruijie router only supports Tail-Drop congestion management instead of RED/WRED congestion management.

Configuring EXP Value of MPLS Message for an Existing Class

This function is optional for CBWFQ. The typical configurations of this function are shown below:

Command	**Function**

Ruijie(config)# **policy-map** *policy-map-name*	Enter/create policy map
Ruijie(config-pmap)# **class** *class-map-name*	Use the class map defined.
Ruijie(config-pmap-c)# **set mpls experimental** *exp-value*	Configure EXP Value of MPLS Message
Ruijie(config-pmap-c)# **set mpls experimental** *dscp*	Set mpls experimental value to ip dscp value
Ruijie(config-pmap-c)# **set mpls experimental** *precedence*	Set mpls experimental value to ip precedence value

Policy-map-name: name of policy map;

Class-map-name: name of class map;

Exp-values: EXP value of message to be configured.

Note

When MPLS experimental value configured is ip dscp, only the first three bits of dscp will be used for mapping.

Configure EXP Value of MPLS Message for an Existing Class (use table-map)

This function is optional for CBWFQ. The typical configurations of this function are shown below:

Command	**Function**
Ruijie(config)# **table-map** *table-map-name*	Enter/create table-map
Ruijie(config-tablemap)#**map from** *from-value* **to** *to-value*	Add mapping relationship into table-map
Ruijie(config-tablemap)#**default** { *default-value* | **copy** | **ignore** }	Specify the action of table-map when specifying unnecessary mapping relationship for table-map
Ruijie(config)# **policy-map** *policy-map-name*	Enter/create policy map
Ruijie(config-pmap)# **class** *class-map--name*	Use the class map defined.
Ruijie(config-pmap-c)# **set mpls experimental dscp table** *table-map-name*	Set mpls experimental value to ip dscp value according to the configuration of table-map
Ruijie(config-pmap-c)# **set mpls experimental precedence table** *table-map-name*	Set mpls experimental value to ip precedence value according to the configuration of table-map
Ruijie(config-pmap-c)# **set mpls experimental qos-group table** *table-map-name*	Set mpls experimental value to qos-group value according to the configuration of table-map

From-value: value mapped;

To-value: map value;

Default-value: default map value;

Table-map-name: name of table map;

Policy-map-name: name of policy map;

Class-map-name: name of class map;

Configuring DSCP Value of IP Message for an Existing Class

This function is optional for CBWFQ. The typical configurations of this function are shown below:

Command	**Function**

Ruijie(config)# **policy-map** *policy-map-name*	Enter/create policy map
Ruijie(config-pmap)# **class** *class-map-name*	Use the class map defined.
Ruijie(config-pmap-c)# **set dscp** dscp-*value*	Configure dscp Value of IP Message
Ruijie(config-pmap-c)# **set dscp experimental**	Set ip dscp to mpls experimental value

Policy-map-name: name of policy map;

Class-map-name: name of class map;

Dscp-values: dscp value of message to be configured.

Note	When MPLS experimental value configured is ip dscp, only the first three bits of dscp will be used for mapping.

Configuring DSCP Value of IP Message for an Existing Class (use table-map)

This function is optional for CBWFQ. The typical configurations of this function are shown below:

Command	**Function**
Ruijie(config)# **table-map** *table-map-name*	Enter/create table-map
Ruijie(config-tablemap)#**map from** *from-value* **to** *to-value*	Add mapping relationship into table-map
Ruijie(config-tablemap)#**default** { *default-value* | **copy** | **ignore** }	Specify the action of table-map when specifying unnecessary mapping relationship for table-map
Ruijie(config)# **policy-map** *policy-map-name*	Enter/create policy map
Ruijie(config-pmap)# **class** *class-map-name*	Use the class map defined.
Ruijie(config-pmap-c)# **set dscp experimental table** *table-map-name*	Set ip dscp to mpls experimental value according to the configuration of table-map
Ruijie(config-pmap-c)# **set dscp qos-group table** *table-map-name*	Set ip dscp to qos-group value according to the configuration of table-map

From-value: value mapped;

To-value: map value;

Default-value: default map value;

Table-map-name: name of table map;

Policy-map-name: name of policy map;

Class-map-name: name of class map;

Configuring Precedence Value of IP Message for an Existing Class

This function is optional for CBWFQ. The typical configurations of this function are shown below:

Command	**Function**
Ruijie(config)# **policy-map** *policy-map-name*	Enter/create policy map
Ruijie(config-pmap)# **class** *class-map-name*	Use the class map defined.

Ruijie(config-pmap-c)# **set precedence** prec-*value*	Configure prec Value of IP Message
Ruijie(config-pmap-c)# **set precedence experimental**	Set ip prec to mpls experimental value

Policy-map-name: name of policy map;

Class-map-name: name of class map;

Prec-values: precedence value of message to be configured.

Configuring Precedence Value of IP Message for an Existing Class (use table-map)

This function is optional for CBWFQ. The typical configurations of this function are shown below:

Command	Function
Ruijie(config)# **table-map** *table-map-name*	Enter/create table-map
Ruijie(config-tablemap)#**map from** *from-value* **to** *to-value*	Add mapping relationship into table-map
Ruijie(config-tablemap)#**default** { *default-value* | **copy** | **ignore** }	Specify the action of table-map when specifying unnecessary mapping relationship for table-map
Ruijie(config)# **policy-map** *policy-map-name*	Enter/create policy map
Ruijie(config-pmap)# **class** *class-map-name*	Use the class map defined.
Ruijie(config-pmap-c)# **set precedence experimental table** *table-map-name*	Set ip prec to mpls experimental value according to the configuration of table-map
Ruijie(config-pmap-c)# **set precedence qos-group table** *table-map-name*	Set ip prec to qos-group value according to the configuration of table-map

From-value: value mapped;

To-value: map value;

Default-value: default map value;

Table-map-name: name of table map;

Policy-map-name: name of policy map;

Class-map-name: name of class map;

Configuring Group ID of Message for an Existing Class

This function is optional for CBWFQ. The typical configurations of this function are shown below:

Command	Function
Ruijie(config)# **policy-map** *policy-map-name*	Enter/create policy map
Ruijie(config-pmap)# **class** *class-map-name*	Use the class map defined.
Ruijie(config-pmap-c)# set qos-group group-*value*	Set the group ID of message
Ruijie(config-pmap-c)# **set qos-group dscp**	Set group ID of message to ip dscp value.
Ruijie(config-pmap-c)# **set qos-group precedence**	Set group ID of message to ip precedence value.
Ruijie(config-pmap-c)# **set qos-group mpls experimental**	Set group ID of message to mpls experimental value.

Ruijie(config-pmap-c)# **set qos-group cos**	Set group ID of message to cos value.

Policy-map-name: name of policy map;

Class-map-name: name of class map;

Group-values: group ID of message to be configured.

Configuring Group ID of Message for an Existing Class (use table-map)

This function is optional for CBWFQ. The typical configurations of this function are shown below:

Command	Function
Ruijie(config)# **table-map** *table-map-name*	Enter/create table-map
Ruijie(config-tablemap)#**map from** *from-value* **to** *to-value*	Add mapping relationship into table-map
Ruijie(config-tablemap)#**default** { *default-value* | **copy** | **ignore** }	Specify the action of table-map when specifying unnecessary mapping relationship for table-map
Ruijie(config)# **policy-map** *policy-map-name*	Enter/create policy map
Ruijie(config-pmap)# **class** *class-map-name*	Use the class map defined.
Ruijie(config-pmap-c)# **set qos-group dscp table** *table-map-name*	Set group ID of message to ip dscp value according to the configuration of table-map
Ruijie(config-pmap-c)# **set qos-group precedence table** *table-map-name*	Set group ID of message to ip precedence value according to the configuration of table-map
Ruijie(config-pmap-c)# **set qos-group mpls experimental table** *table-map-name*	Set group ID of message to mpls experimental value according to the configuration of table-map
Ruijie(config-pmap-c)# **set qos-group cos table** *table-map-name*	Set group ID of message to cos value according to the configuration of table-map

From-value: value mapped;

To-value: map value;

Default-value: default map value;

Table-map-name: name of table map;

Policy-map-name: name of policy map;

Class-map-name: name of class map;

Configuring the Bandwidth Allocated to CBWFQ

This function is optional for CBWFQ. The typical configurations of this function are shown below:

Command	Function
Ruijie(config-if)# **max-reserved-bandwidth** *percent*	Configure the Bandwidth Allocated to CBWFQ

Percent: Percentage of bandwidth allocated to CBWFQ on the existing network interface.

The user may allocate the available bandwidth percentage allocated to CBWFQ. The default value is 75, which means 75% of gross available bandwidth of network interface will be allocated to CBWFQ.

Monitoring CBWFQ

When CBWFQ becomes valid on a specific interface, to display input and output queues, the user can execute the following commands in privileged user mode:

Command	**Function**
Ruijie# **show class-map**	Display all class maps
Ruijie# **show class-map** *class-map-name*	Display the information of a specific class map
Ruijie# **show policy-map**	Display all policy maps
Ruijie# **show policy-map name** *policy-map –name*	Display the information of a specific policy map
Ruijie# **show policy-map name** *policy-map-name* class *class-name*	Display a specific class map in a specific policy map
Ruijie# **show policy-map interface** *Interface-name interface-number*	Display the policy map applied on a specific network interface
Ruijie# **show queue** *interface-name interface-number*	Display the QoS queue information on the designated network interface.

Class-map-name: name of class map;

Policy-map-name: name of policy map;

Interface-name: name of network interface;

Interface-number: network interface ID.

CBWFQ Configuration Examples

The following example shows how to configure MPLS Experimental based CBWFQ congestion management policy on the synchronization interface:

```
class-map 101
match mpls experimental 1
class-map 102
match mpls experimental 2
class-map 103
match mpls experimental 3
!
policy-map 1
 class 101
  bandwidth 600
 class 102
  bandwidth 400
 class 103
  bandwidth 200
!
interface Serial 3/0
 ip ref
 ip address 192.168.200.1 255.255.255.0
 mpls ip
 service-policy output 1
```

An example of viewing interface configurations in privileged user mode is shown below:

```
Ruijie# show queue interface serial 3/0

  Queueing strategy: cb weighted fair
  Output queue: 0/300/128/0 (size/max total/threshold/drops)
   cb queue_num  0/0 (active/max active)
   wfq queue_num  0/0 (active/max active)
     Reserved queue_num 3/3 (allocated/max allocated)
     Llq is close

  Qos Ref queue information
  Current Policy(s) : CBWFQ
  interface cir: 2048000
  Queueing strategy: cb weighted fair
   Dequeue threshold:  Green 25000, Yellow 37500, Red 50000
```

```
    Queues: Queues total len 0, MeanBurst 800
    Queues: gts gap 7, deta bits 262, token bucket 51200
    Queues: Max 19353 pkts, used 0 pkts
    Queues: rtpQ: 0 pkts, 0 bytes
    Queues: llQ:  0 pkts, 0 bytes
    Queues: genQ: 0 pkts, 0 bytes
   Output Stat.: 0/0/ (send/drops)
    queue_num  128/256 (cb num/wfq num)
    cb packet Stat. 0/0 (send /drop)
    wfq packet Stat. 0/0 (send /drop)
      Reserved queue_num 3/3 (allocated/max allocated)
      Llq is close
```

Configuration of Custom Queueing (CQ)

CQ Configuration Tasks

Custom Queue configuration tasks are shown below:

Configuring CQ

CQ can configure up to 16 groups, namely the scope of List-number is 1-16. Each group specifies the type of queues that can be entered by different kinds of packets, queue length, and byte count that is allowed to be sent.

Determining the Maximum Capacity of Queue Adopting CQ

To configure the maximum packet capacity for each queue, execute the following commands in global configuration mode:

Command	Function
Ruijie(config)# **queue-list** *list-number* **queue** *queue-number* **limit** *limit-number*	Specify the maximum number of packets allowed by each custom queue. The no form of this command can be used to restore the queue length to the default value of 20.
Ruijie(config)# **queue-list** *list-number* **queue** *queue-number* **byte-count** *byte-count-number*	Specify the number of bytes allowed by each queue. The no form of this command can be used to restore the byte count to the default value of 1500.

List-number: number of queue list (any number between 1-16);

Queue-number: queue number (any number between 1-16);

Limit-number: The maximum number of packets allowed by the queue, within the range being from 1 to 32767. The default value is 20.

Byte-count-number: Specify how many bytes of data should be delivered from the current queue by the system before the system moves on to the next queue. When a particular queue is being processed, packets are sent until the number of bytes sent exceeds the byte-count-number configured (within the range of 1 to 16777215, 1500 by default) or until the queue is empty. For how to determine the best byte count of data to be sent, please refer to the foregoing chapters.

Note

To configure CQ congestion management policy on the interface, all interfaces of the system must disable the fast forwarding function, which is not supported by CQ congestion management policy.

Assigning Packets to CQ

You can assign the packets to custom queues based on the protocol type or the interface where the packets enter the device. Additionally, you can set the default queue for the packets that do not match other assignment rules. You can also specify multiple rules.

To define the CQ lists, use the following commands in global configuration mode:

Command	Function

Ruijie(config)# **queue-list** *list-number* **protocol mpls** *queue-number* [**experimental** *exp-value*]	Assign the packets to the specified custom queue based on the protocol type.

Therein, exp-value refers to MPLS experimental value.

Applying CQ List on the Interface

To apply a CQ list to an interface, use the following command in interface configuration mode:

Command	**Function**
Ruijie(config-if)# **custom-queue-list** *list-number*	Set the queuing policy of this interface to a specific CQ list.

Note	You can only specify one queuing policy for each interface, and only one queue list can be specified in the mean time.

Monitoring CQ

To display information about input and output queues when CQ is enabled on an interface, use the following commands in privileged user mode:

Command	**Function**
Ruijie# show queue cq	Display information about CQ.
Ruijie# **show queue interface** *interface-name interface-number* [*queue-number*]	Display information about CQ interface.
Ruijie# **debug qos cq**	Turn on the CQ debug switch if the priority queue is configured.

CQ Configuration Examples

Configure custom list 2, and assign packets with MPLS EXP value being 2 to queue number 12 :

```
Ruijie(config)# queue-list 2 protocol mpls 12 experimental 2
```

Configure custom list 1, and assign packets with MPLS EXP value being 5 to queue number 11:

```
Ruijie(config)# queue-list 1 protocol mpls 11 experimental 5
```

Apply custom list 1 configured previously to the synchronization interface:

```
Ruijie(config)# interface serial 0
Ruijie(config-if)# custom-queue-list 1
```

Configuration of Priority Queueing (PQ)

PQ Configuration Tasks

Priority Queue configuration tasks are shown below:

Configuring PQ

PQ can configure up to 16 groups, namely the scope of List-number is 1-16. Each group specifies the type of queues that can be entered by different kinds of packets, as well as the maximum number of packets allowed by each queue.

Determining the Maximum Capacity of Queue Adopting PQ

In the queue list of each group, there are four queues divided into high, medium, normal and low. The user may configure the maximum packet capacity of each queue. Execute the following commands in global configuration mode:

Command	Function
Ruijie(config)# **priority-list** *list-number* **queue-limit** *high-limit medium-limit normal-limit low-limit*	Specify the maximum number of packets allowed by each priority queue.

List-number: number of queue list (any number between 1-16);

The default length of priority queue is shown below:

Queue	Length
high	20
medium	40
normal	60
Low	80

Note To configure PQ congestion management policy on the interface, all interfaces of the system must enable fast forwarding function, otherwise this function will become invalid.

Assigning Packets to PQ

The system can specify multiple assignment rules. This list will be searched according to the sequence specified by priority-list until a matching protocol or interface type is found. When a matching entry is found, this packet will be allocated to the corresponding queue and the search will end. Packets failing to match other assignment rules can be allocated to the default queue. To specify the queue for assigning packets, execute the following command in global configuration mode:

Command	Function
Ruijie(config)# **priority-list** *list-number* **protocol mpls {high | medium | normal | low}** [**experiemtal** *exp-value*]	Assign packets to a specific PQ according to the EXP value of MPLS message.
Ruijie(config)# **priority-list** *list-number* **default {high | medium | normal | low}**	Assign packets matching no rules to the default PQ (normal).

Therein, List-number is the group ID of PQ, and exp-value refers to MPLS experimental value.

Applying PQ list on the Interface

To apply a PQ list to an interface, execute the following command in interface configuration mode:

Command	Function
Ruijie(config-if)# priority-group *list-number*	Set the queuing policy of this interface to a specific PQ list.

Note You can only specify one queuing policy for each interface, and only one queue list can be specified in the mean time.

Monitoring PQ

When PQ becomes valid on a specific interface, to display input and output queues, the user can execute the following commands in privileged user mode:

Command	Function

Ruijie# **show queue interface** *interface–name interface-number [queue-number]*	Display the PQ information on the designated network interface.

PQ Configuration Examples

Configure priority list 1, and allocate packets with MPLS EXP value being 7 to the medium PQ:

```
priority-list 1 mpls experiemtal 7 medium
```

Configure priority list 1, and allocate packets received by synchronization serial port to the medium PQ:

```
priority-list 1 interface serial 1/1 medium
```

Configure priority list 1, and allocate packets matching no rules in the priority list to the medium PQ:

```
priority-list 1 default medium
```

Configure priority list 1, and configure the length of high, medium, normal and low PQs to 10, 40, 60 and 80 respectively:

```
priority-list 1 queue-limit 10 40 60 80
```

Apply priority list 1 configured previously to the synchronization interface:

```
interface Serial 3/0
```

```
 ip ref
 ip address 192.168.200.1 255.255.255.0
 mpls ip
 priority-group 1
```

An example of viewing interface configurations in privileged user mode is shown below:

```
Ruijie# show queue interface serial 3/0

  Queueing strategy: priority-list 1
  Output queues: (queue #: size/max/send/drops)
  Output queue: high 0/20/0/0, medium 0/40/0/0, normal 0/60/0/0, low 0/80/0/0

  Qos Ref queue information
  Current Policy(s) : PQ
  Queueing strategy: priority-list 1
  interface cir: 2048000
  Dequeue threshold:  Green 25000, Yellow 37500, Red 50000
  Queues: Queues total len 0, MeanBurst 800
  Queues: gts gap 7, deta bits 262, token bucket 51200
  Queues: Max 19353 pkts, used 0 pkts
  Queues: rtpQ: 0 pkts, 0 bytes
  Queues: llQ:  0 pkts, 0 bytes
  Queues: genQ: 0 pkts, 0 bytes
  Threshold: MeanBurst 800, Priority LOW, Dec 4560, Inc 4560, Drop 16720
  Counter:   PriInc 0, PriDec 0, Drop 0
  Queues: Queues len 0, MeanBurst 800, gts token bucket 51200
  Queues: Max 19353 pkts, used 0 pkts, rtpQ: 0 pkts, 0 bytes. genQ: 0 pkts, 0 bytes
  (size/max/send/drops)
  high 0/0/0/0, medium 0/0/0/0, normal 0/0/0/0, low 0/0/0/0
```

Configuration of Low Latency Queuing (LLQ)

LLQ Configuration Tasks

LLQ configuration is performed jointly with CBWFQ configuration. To configure LLQ, the following tasks need to be done:

Configuring LLQ

To configure LLQ, execute the following commands in Policy-map command layer configuration mode:

Command	**Function**
Ruijie(config)# **policy-map** *policy-map-name*	Enter/create policy map
Ruijie(config-pmap)# **class** *class-map-name*	Use the class map defined.
Ruijie (config-pmap-c)# **priority** { *bandwidth-kbps* | **percent** *percent*} [*Burst bytes*]	Allocate bandwidth for specific class of traffic

Ruijie(config-if)# **service-policy output** *policy-map-name*	Enable CBWFQ and specify the policy map to be applied.

Policy-map-name: name of policy map;

Class-map-name: name of class map;

Bandwidth-kbps: bandwidth allocated (unit: kbps);

Percent: percentage of bandwidth allocated (in regard to all available bandwidth of network interface);

The user may allocate bandwidth for the specific type of network traffic. By default, 1% of bandwidth is allocated to the specific type of network traffic.

Burst bytes: number of bytes allowed in a burst above the committed rate limit.

Note

Generally, the bandwidth allocation on the interface will be influenced by the following commands: Bandwidth (CBWFQ), Priority (LLQ), ip rtp priority (RTPQ), and max-reserved-bandwidth (interface). The total bandwidth depends on the "Bandwidth" command of interface. I.e., Bandwidth = max-reserved-bandwidth +default wfq bandwidth; max-reserved-bandwidth = bandwidth(policy-map) +priority(policy-map) +ip RTP priority

Note

To configure PQ congestion management policy on the interface, all interfaces of the system must enable fast forwarding function, otherwise this function will become invalid.

Monitoring LLQ

To view LLQ configurations of interface, execute the following command in privileged user mode.

Command	**Function**
Ruijie# **show policy-map** *interface-name interface-number*	Display the interface information of LLQ.

LLQ Configuration Examples

The following example shows how to configure a LLQ on the synchronization interface to serve packets with MPLS EXP value being 7:

```
class-map match-all 201
 match mpls experimental 7
!
policy-map 1
 class 201
  priority 30 2000
!
interface Serial 3/0
 ip ref
 ip address 192.168.200.1 255.255.255.0
 mpls ip
 service-policy output 1
```

An example of viewing interface configurations in privileged user mode is shown below:

```
Ruijie#show queue interface serial 3/0

  Queueing strategy: cb weighted fair
  Output queue: 0/300/128/0 (size/max total/threshold/drops)
   cb queue_num  0/0 (active/max active)
   wfq queue_num  0/0 (active/max active)
     Reserved queue_num 1/1 (allocated/max allocated)
     Llq is open

  Qos Ref queue information
  Current Policy(s) : CBWFQ
```

```
 interface cir: 2048000
 Queueing strategy: cb weighted fair
  Dequeue threshold:  Green 25000, Yellow 37500, Red 50000
   Queues: Queues total len 0, MeanBurst 800
   Queues: gts gap 7, deta bits 262, token bucket 51200
   Queues: Max 19353 pkts, used 0 pkts
   Queues: rtpQ: 0 pkts, 0 bytes
   Queues: llQ:  0 pkts, 0 bytes
   Queues: genQ: 0 pkts, 0 bytes
  Output Stat.: 0/0/ (send/drops)
   queue_num  128/256 (cb num/wfq num)
   cb packet Stat. 0/0 (send /drop)
   wfq packet Stat. 0/0 (send /drop)
     Reserved queue_num 1/1 (allocated/max allocated)
     Llq is open
```

Traffic Policing and Traffic Shaping

Introduction to Traffic Policing and Traffic Shaping

In traffic policing, certain actions will be taken to limit the data rate of classified traffics entering the network.

Traffic shaping will restrict the burst of traffics, so that message flows can be transmitted at an even rate and the network traffic can maintain stable.

Traffic Policing Configuration Tasks

To configure Car traffic shaping on the interface, execute the following commands in interface configuration mode:

Command	**Function**
Ruijie(config)# **interface** *interface-type* *interface-number*	Specify the interface for Car traffic policing.
Ruijie(config-if)# **rate-limit {input | output} bps** *burst-normal burst-max* **conform-action** *action* **exceed-action** *action*	Applying rate limiting on all traffics entering the receiving interface or outgoing interface.

Input|output: The input or output data rate to be limited by the user.

Bps: Maximum data rate of the traffic desired by the user (unit: bps).

Burst-normal burst-max: Size of token bucket (unit: bytes).

Conform-action: Traffic handling policy under rate limitation.

Exceed-action: Data rate handling policy when rate limit is exceeded.

Action: Handling policy, including:

Continue to match the next policy

- Drop: drop the packet
- Set-mpls-exp-transmit: transmit this packet after setting mpls experimental field
- Set-mpls-exp-continue: after setting mpls experimental field, this packet continue to match the next policy
- Transmit: transmit this packet

Note To configure traffic policing on the interface, all interfaces of the system must enable fast forwarding function, otherwise this function will become invalid.

Traffic Shaping Configuration Tasks

To configure GTS traffic shaping on the interface, execute the following commands in the interface configuration mode:

Command	**Function**

Ruijie(config)# **interface** *interface-type* *interface-number*	Specify the interface for traffic shaping.
Ruijie(config-if)# **traffic-shape rate** *bit-rate* [*burst-size*] [*excess-burst-size*] [*buffer-limit*]	Carry out traffic shaping of all traffics on the interface.

Bit-rate: the maximum data rate to be shaped by the user (unit: bps).

Burst-size: the maximum traffic size that is permitted in each burst at each interval (unit: bit)

Excess-size: transient burst of traffic that the first interval can forward (unit: bit)

Buffer-limit: buffer size of gts buffer queue (default: 1000).

Note

To configure traffic shaping on the interface, all interfaces of the system must enable fast forwarding function, otherwise this function will become invalid.

Note

The traffic shaping policy handled by the system will function on the interface. When GTS has been configured for the interface, all related sub-interfaces of this interface must enable GTS, otherwise the traffic forwarding will become uneven on related sub-interfaces.

Note

After traffic shaping is enabled on the interface, the burst traffic must be the integral multiple of the data transmitted at 10ms under traffic-shaping rate, otherwise the system will round off the burst traffic configuration parameters according to the data transmitted at 10ms under traffic-shaping rate, so that the parameters can become valid.

Configuring Traffic Policing under Policy-map

To configure single-rate Car traffic limiting in Policy-map, execute the following commands:

Command	**Function**
Ruijie(config)# **policy-map** *policy-map-name*	Enter/create policy map
Ruijie(config-pmap)# **class** *class-map-name*	Use the class map defined.
Ruijie(config-pmap-c)# **police cir** *bps burst-normal burst-max* **conform-action** *action* **exceed-action** *action* **violate-action** *action*	Perform single-rate token bucket limiting of such traffics.
Ruijie(config-if)# **service-policy output** *policy-map-name*	Specify the policy map to be applied to the interface.

CIR: Maximum data rate of the traffic desired by the user (unit: CIR).

Burst-normal burst-max: Size of token bucket (unit: bytes).

Conform-action: Traffic handling policy under rate limitation.

Exceed-action: Data rate handling policy when rate limit is exceeded.

Violate-action: Traffic handling policy when the second token bucket rate limit is exceeded in the case of two token bucket system.

Action: Handling policy, including:

- Drop: drop the packet
- Set-mpls-exp-transmit: transmit this packet after setting mpls experimental field
- Transmit: transmit this packet

To configure dual-rate Car traffic limiting in Policy-map, execute the following commands:

Command	**Function**
Ruijie(config)# **policy-map** *policy-map-name*	Enter/create policy map
Ruijie(config-pmap)# **class** *class-map-name*	Use the class map defined.
Ruijie(config-pmap-c)# **police cir** *bps* **pir** *bps* *burst-normal burst-max* **conform-action** *action* **exceed-action** *action* **violate-action** *action*	Perform dual-rate token bucket limiting of such traffics.
Ruijie(config-if)# **service-policy output** *policy-map-name*	Specify the policy map to be applied to the interface.

CIR: Maximum data rate of the traffic desired by the user (unit: CIR).

PIR: Peak data rate of the traffic desired by the user (unit: CIR).

Burst-normal burst-max: Size of token bucket (unit: bytes).

Conform-action: Traffic handling policy under rate limitation.

Exceed-action: Data rate handling policy when rate limit is exceeded.

Violate-action: Traffic handling policy: when the second token bucket rate limit is exceeded in the case of two token bucket system.

Action: Handling policy, including:

- Drop: drop the packet
- Set-mpls-exp-transmit: transmit this packet after setting mpls experimental field
- Transmit: transmit this packet

Note There are four token bucket algorithms for rate limiting under policy-map. The user may select different token bucket algorithm according to different configurations.

Note Single token bucket algorithm: If violate-action is not configured and the value of burst-normal equals to the value of burst-max, the single token bucket algorithm is adopted.

Note Borrowing mode of single token bucket algorithm: If violate-action is not configured and the value of burst-normal is smaller than the value of burst-max, the borrowing mode of single bucket algorithm is adopted.

Note Single-rate token bucket algorithm: If violate-action is configured but pir is not configured, the single-rate dual token bucket algorithm is adopted.

Note Dual-rate token bucket algorithm: If both violate-action and pir are configured, the dual-rate dual token bucket algorithm is adopted.

Traffic Policing Configuration Examples

Example of Applying Traffic Policing on All Traffics on the Interface

# Configure car traffic policing of incoming messages on Serial interface

# Limit traffics on the receiving interface at 2Mbps; transmit conforming traffics after setting mpls experimental value to 2 and drop excess traffics.

```
interface Serial 3/0
 ip ref
 ip address 192.168.20.3 255.255.255.0
 mpls ip
 rate-limit input 2000000 3000 3000 conform-action set-mpls-exp-transmit 2 exceed-action  drop
```

An example of viewing interface configurations in privileged user mode is shown below:

```
Ruijie#show rate-limit interface serial 3/0
Serial 3/0
  Input
    matches all traffic
      params:  2000000 bps, 3000 limit, 3000 extended limit
      conformed 0 packets, 0 bytes; action: set mpls transmit
      exceeded 0 packets, 0 bytes; action: drop
      cbucket 6000, cbs 6000; ebucket 0 ebs 0
```

Example of Traffic Policing Configuration under Policy-map

The following example shows how to apply Policy-map based traffic limiting on conforming traffics on the outgoing interface. Single-rate dual token bucket algorithm is applied to limit each kind of traffic.

```
!
class-map match-all a1
match mpls experimental 1
class-map match-all a2
match mpls experimental 2
class-map match-all a3
match mpls experimental 3
class-map match-all a4
match mpls experimental 4
!
policy-map police
class a1
police cir 80000 2000 2000 conform-action transmit exceed-action drop violate-action drop
class a2
police cir 160000 2000 2000 conform-action transmit exceed-action drop violate-action drop
class a3
police cir 320000 6000 6000 conform-action transmit exceed-action drop violate-action drop
class a4
police cir 640000 6000 6000 conform-action transmit exceed-action drop violate-action drop
!
interface Serial 3/0
 ip ref
 ip address 192.168.20.3 255.255.255.0
 mpls ip
 service-policy output police
```

# View interface configurations in privileged user mode:

```
Ruijie#show policy-map interface serial 3/0

Serial 3/0  output(tc policy): police
    Class a1
      current token tbf: TC_SRTMC
      params:  80000 bps, 2000 limit, 2000 extended limit , 0 pir
      conformed 0 packets, 0 bytes; action: transmit 0
      exceeded 0 packets, 0 bytes; action: drop 0
      violated 0 packets, 0 bytes; action: drop 0
      cbucket 2000, cbs 2000; ebucket 2000 ebs 2000
    Class a2
      current token tbf: TC_SRTMC
      params:  160000 bps, 2000 limit, 2000 extended limit , 0 pir
      conformed 0 packets, 0 bytes; action: transmit 0
      exceeded 0 packets, 0 bytes; action: drop 0
      violated 0 packets, 0 bytes; action: drop 0
      cbucket 2000, cbs 2000; ebucket 2000 ebs 2000
    Class a3
      current token tbf: TC_SRTMC
```

```
     params:  320000 bps, 6000 limit, 6000 extended limit , 0 pir
     conformed 0 packets, 0 bytes; action: transmit 0
     exceeded 0 packets, 0 bytes; action: drop 0
     violated 0 packets, 0 bytes; action: drop 0
     cbucket 6000, cbs 6000; ebucket 6000 ebs 6000
   Class a4
     current token tbf: TC_SRTMC
     params:  640000 bps, 6000 limit, 6000 extended limit , 0 pir
     conformed 0 packets, 0 bytes; action: transmit 0
     exceeded 0 packets, 0 bytes; action: drop 0
     violated 0 packets, 0 bytes; action: drop 0
     cbucket 6000, cbs 6000; ebucket 6000 ebs 6000
```

The following example shows how to apply Policy-map based traffic limiting on conforming traffics on the outgoing interface. Dual-rate dual token bucket algorithm is applied to limit each kind of traffic.

```
!
policy-map police
class a1
police cir 80000 pir 100000 2000 2000 conform-action transmit exceed-action drop violate-action
drop
class a2
police cir 160000 pir 200000 2000 2000 conform-action transmit exceed-action drop
violate-action drop
class a3
police cir 320000 pir 400000 6000 6000 conform-action transmit exceed-action drop
violate-action drop
class a4
police cir 640000 pir 700000 6000 6000 conform-action transmit exceed-action drop
violate-action drop
!
interface Serial 3/0
 ip ref
 ip address 192.168.20.3 255.255.255.0
 mpls ip
 service-policy output police
```

# View interface configurations in privileged user mode:

```
Ruijie#show policy-map interface serial 3/0

Serial 3/0  output(tc policy): police
   Class a1
     current token tbf: TC_TRTMC
     params:  80000 bps, 2000 limit, 2000 extended limit , 100000 pir
     conformed 0 packets, 0 bytes; action: transmit 0
     exceeded 0 packets, 0 bytes; action: drop 0
     violated 0 packets, 0 bytes; action: drop 0
     cbucket 2000, cbs 2000; ebucket 2000 ebs 2000
   Class a2
     current token tbf: TC_TRTMC
     params:  160000 bps, 2000 limit, 2000 extended limit , 200000 pir
     conformed 0 packets, 0 bytes; action: transmit 0
     exceeded 0 packets, 0 bytes; action: drop 0
     violated 0 packets, 0 bytes; action: drop 0
     cbucket 2000, cbs 2000; ebucket 2000 ebs 2000
   Class a3
     current token tbf: TC_TRTMC
     params:  320000 bps, 6000 limit, 6000 extended limit , 400000 pir
     conformed 0 packets, 0 bytes; action: transmit 0
     exceeded 0 packets, 0 bytes; action: drop 0
     violated 0 packets, 0 bytes; action: drop 0
     cbucket 6000, cbs 6000; ebucket 6000 ebs 6000
   Class a4
     current token tbf: TC_TRTMC
     params:  640000 bps, 6000 limit, 6000 extended limit , 700000 pir
     conformed 0 packets, 0 bytes; action: transmit 0
     exceeded 0 packets, 0 bytes; action: drop 0
     violated 0 packets, 0 bytes; action: drop 0
     cbucket 6000, cbs 6000; ebucket 6000 ebs 6000
```

Traffic Shaping Configuration Examples

Example of Applying Traffic Shaping on All Traffics on the Interface

# Configure GTS traffic shaping of outgoing messages on Serial interface

# Shape traffics on the outgoing interface at 300kbps; transmit conforming traffics and put excess traffics in the buffer queue for later transmission.

```
interface Serial 3/0
 ip ref
 ip address 192.168.20.3 255.255.255.0
 mpls ip
 traffic-shape rate 2000000 40000 40000 1000
# View interface configurations in privileged user mode:
Ruijie#show traffic-shape serial 3/0
Interface Serial 3/0
      Access Target   Byte     Sustain   Excess    Interval  Increment Adapt
VC    List   Rate     Limit    bits/int  bits/int  (ms)      (bytes)   Active
-     -     2000000   10000    40000    40000    20       5000     -
```

Congestion Avoidance

Introduction to Congestion Avoidance

Congestion avoidance is deployed at the network bottle neck to effectively monitor network traffics and avoid anticipated congestion developed at the network bottle neck by dropping information packets.

Excess congestion will cause great harms to network resources, and certain measures shall be taken accordingly. The Congestion Avoidance as mentioned herein refers to the mechanism of actively dropping packets when congestion is expected by monitoring how the network resources are utilized (such as queues or memory buffers), as so to alleviate the load on the network.

WRED

WRED avoids global TCP synchronization by randomly dropping packets. Thus, while the sending rates of some TCP sessions slow down after their packets are dropped, other TCP sessions remain at high sending rates. As there are always TCP sessions at high sending rates, link bandwidth is efficiently utilized.

Before dropping packets, WRED will compare the queue size with lower threshold and upper threshold (the absolute length of queue threshold), and this will result in the unequal treatment of burst traffics and compromise traffic transmission. Therefore, when dropping packets by comparing between lower threshold and upper threshold, the average size of queue will be adopted (this should be the relative value upon comparison between queue threshold and average size). The average queue size is the result of low pass filtering of queue size, and avoids the unequal treatment in the burst of queue size as it reflects the change tendency of queue and is not sensitive to the changes in queue size. The relationship between WRED and queuing mechanism is shown below:

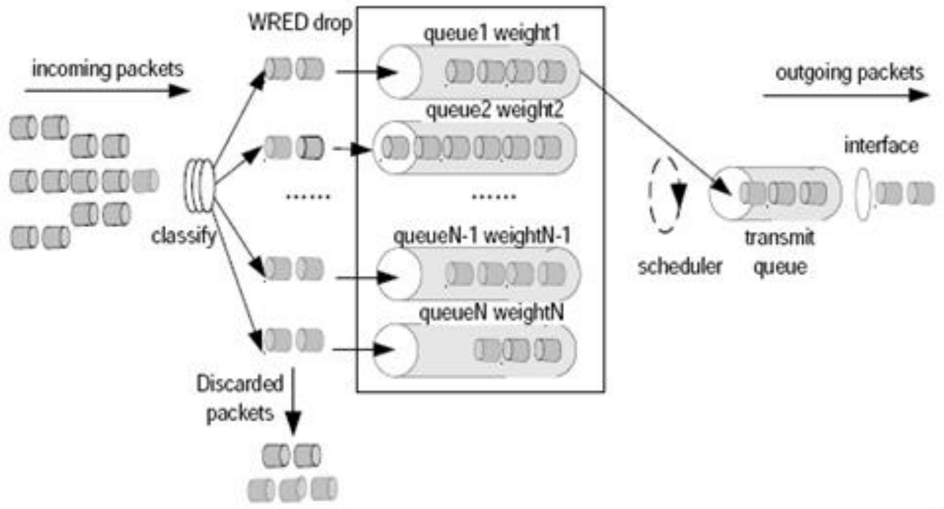


Figure 1

Upper and lower thresholds are set for each queue, and the packets in a queue are processed as follows:

- When the queue size is shorter than the lower threshold, no packet is dropped;
- When the queue size reaches the upper threshold, all subsequent packets are dropped;
- When the queue size is between the lower threshold and the upper threshold, the WRED algorithm will be adopted to determine whether the packets will be dropped or not.

In practice, a random number will be assigned to each incoming packets, and this random number is compared with the drop probability of the existing queue. If this random number is larger than the drop probability, the packets will be dropped. The longer the queue is, the higher the drop probability will be. However, there will be a maximum drop probability.

Configure Congestion Avoidance

Congestion Avoidance (WRED) Configuration Tasks

To enable MPLS EXP value based congestion avoidance on the interface, execute the following command:

Command	Function
Ruijie(config-if)# **random-detect mpls-exp-based**	Enable congestion avoidance based on the EXP value of MPLS packets.

Note

To configure congestion avoidance on the interface, all interfaces of the system must enable fast forwarding function, otherwise this function will become invalid.

To configure the maximum threshold, minimum threshold and drop probability of each kind of traffics classified by experimental, execute the following commands in interface configuration mode:

Command	Function
Ruijie(config)# **interface** *interface-type interface-number*	Specify the interface for congestion avoidance.
Ruijie(config-if)# **random-detect experimental** *exp-value min-threshold max-threshold mark-prob-denominator*	Configure the maximum threshold, minimum threshold and drop probability of each kind of traffics classified by experimental.

Exp-value: experimental value; traffics are classified according to this value.

Min-threshold: the minimum drop threshold; the default value differs from traffic to traffic.

Max-threshold: the maximum drop threshold; the default value differs from traffic to traffic.

Mark-prob-denominator: drop probability; the default value is 10, i.e., 1/10. The larger this value is, the smaller the drop probability will be.

To configure the weight factor for traffic congestion avoidance on the interface, execute the following commands in privileged user mode:

Command	Function
Ruijie(config)# **interface** *interface-type* *interface-number*	Specify the interface for congestion avoidance.
Ruijie(config-if)# **random-detect** **exponential-weighting-constant** *exponential-value*	Configure the weight factor for traffic congestion avoidance on the interface.

Exponential-value: the default value of weight factor is 9; the smaller this value is, the larger the drop probability will be; the larger this value is, the smaller the drop probability will be.

Note

If the queuing algorithm of Ethernet interface is not FIFO, you must "no" the queuing algorithm before configuring WRED congestion avoidance on the interface. If the queuing algorithm of synchronization interface is not FIFO or WFQ, you must "no" the queuing algorithm before configuring WRED congestion avoidance on the interface.

Congestion Avoidance (WRED) Configuration Examples

Example of Configuring Congestion Avoidance on the Interface

# Configure WRED congestion avoidance based on MPLS experimental traffic classification on the synchronization interface.

# Set lower threshold to 5, upper threshold to 100 and drop probability to 10 for packets with experimental value being 1.

# Set lower threshold to 10, upper threshold to 100 and drop probability to 10 for packets with experimental value being 2.

# Set lower threshold to 20, upper threshold to 100 and drop probability to 10 for packets with experimental value being 3.

# Set lower threshold to 30, upper threshold to 100 and drop probability to 10 for packets with experimental value being 4.

```
interface Serial 3/0
```

```
 ip ref
 ip address 192.168.20.3 255.255.255.0
 mpls ip
 random-detect mpls-exp-based
 random-detect experimental 1 5 100 10
 random-detect experimental 2 10 100 10
 random-detect experimental 3 20 100 10
 random-detect experimental 4 30 100 10
# View interface configurations in privileged user mode:
Ruijie#show queue interface serial 3/0

Current random-detect configuration:
   Serial 3/0
      Queueing strategy: random early detection (WRED)
      Exp-weight-constant: 9 (1/512)
      Mean queue depth: 0
class          Random drop     Tail drop    Minimum Maximum  Mark
               pkts/bytes      pkts/bytes    thresh  thresh  prob
0        0/0              0/0             20      40    1/10
1        0/0              0/0              5     100    1/10
2        0/0              0/0             10     100    1/10
3        0/0              0/0             20     100    1/10
4        0/0              0/0             30     100    1/10
5        0/0              0/0             31      40    1/10
6        0/0              0/0             33      40    1/10
7        0/0              0/0             35      40    1/10

  Qos Ref queue information
  Current Policy(s) : WRED
  Queueing strategy: random early detection (WRED)
  interface cir: 2048000
  Dequeue threshold:  Green 25000, Yellow 37500, Red 50000
  Queues: Queues total len 0, MeanBurst 800
  Queues: gts gap 7, deta bits 262, token bucket 51200
  Queues: Max 19353 pkts, used 0 pkts
  Queues: rtpQ: 0 pkts, 0 bytes
  Queues: llQ:  0 pkts, 0 bytes
  Queues: genQ: 0 pkts, 0 bytes
```

QoS Overview

Understanding QoS

QoS Overview

Devices on a conventional IP network equally treat all data packets with a First In First Out (FIFO) policy and deliver each data packet with the best effort to the destination. They do not provide any guarantee for packet transmission performance such as reliability and transmission delay.

As the Internet becomes rapidly popular in the globe and information networks emerge one after another in today's society, people raise increasingly-higher requirements for networks. Today, information requirements are no longer confined to mere data information but also extend to interactive multimedia. Services are developing towards data, voice, unified image, and integrated network transmission. Highly-real-time voice, image, and important data services sensitive to bandwidth delay and jitter tend to be more widely transmitted on networks. On one hand, this helps greatly improve network resources. On the other hand, an issue about how to guarantee network Quality of Service (QoS) arises, since voice, data, and image services have different delay, throughput, and packet loss rate requirements.

The QoS mechanism is intended to provide different QoS to meet diversified service quality requirements.

Basic Concepts

Three QoS models are defined to meet different service quality requirements. They are the Best-Effort Service model, the Integrated Service Model, and the Differentiated Service (DiffServ) model.

Best-Effort Service

The Best-Effort Service model enables a network to transmit packets with the best effort but does not provide any guarantee for transmission performance such as delay and reliability. It is a default service model applied on the conventional Internet.

Integrated Service

In the Integrated Service model, an application program needs to submit a specific QoS request to the network before sending a packet. The request covers the required bandwidth and delay. The application program starts to send the packet only after receiving an acknowledgment from the network (that is, after the network reserves resources for the application program). In addition, packets sent by the application program must be controlled within the traffic range described by traffic parameters.

Differentiated Service

In the Differentiated Service model, an application program does not need to submit a resource request to the network before sending a packet. Instead, it sets QoS parameter information in the IP header of the packet to inform network nodes of its QoS requirements. All routers on the packet propagation path can analyze the IP header to obtain the QoS class of the packet.

Working Principles

Major QoS technologies include traffic classification, traffic policing, traffic shaping, congestion management, and congestion avoidance. The Integrated Service model also introduces a protocol called the Resource Reservation Protocol (RSVP).

Traffic Classification

Objects are identified based on certain matching rules. Traffic classification is a precondition for implementing differentiated services.

Traffic Policing

Traffic policing is used to monitor and control the specifications of specific traffic inbound to routers. It applies when traffic goes beyond specifications.

Traffic Shaping

Traffic shaping is a means to control traffic by actively adjusting the output rate of traffic. In general, it enables traffic to adapt to available network resources on the downstream router so as to avoid packet loss or congestion.

Congestion Management

Congestion management is a measure for solving resource contention in the event of network congestion. In general, packets are cached in queues and a certain scheduling algorithm is applied to arrange the forwarding sequence of packets.

Congestion Avoidance

Excessive congestion will cause great harm to network resources.

RSVP

RSVP is an end-to-end (E2E) resource reservation protocol. Resource requests are transmitted between network nodes. Upon receipt of these requests, a network node needs to allocate resources for these requests.

Ruijie QoS mechanism supports the DiffServ model. The following sections describe how to configure QoS technologies.

Traffic Classification Configuration

Understanding Traffic Classification

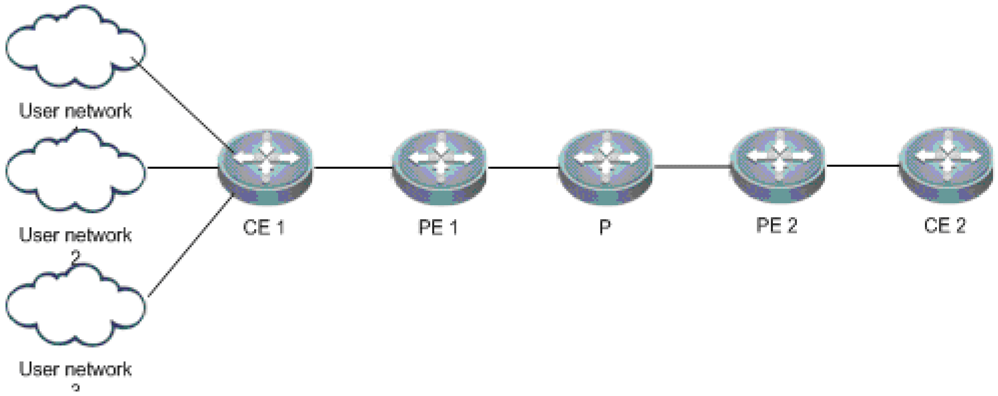
Traffic Classification Overview

When the DiffServ model is applied for QoS implementation, routers need to identify various flows and therefore traffic classification must be performed for packets. Two methods can be used for traffic classification: complex traffic classification and simple traffic classification.

Complex traffic classification is a means to classify packets in a fine manner using complex rules, such as rules based on link layer, network layer, and transport layer information (e.g. source MAC address, destination MAC address, source IP address, destination IP address, user group number, protocol type, or TCP/UDP port number of applications). In general, complex traffic classification is applied to traffic on border routers in the DiffServ domain.

As shown in Figure 9, CE 1, PE 1, P, PE 2, and CE 2 form a service provider network. PE 1, P, and PE 2 establish MPLS neighbors with each other. User networks 1, 2, and 3 access an MPLS network from CE 1 which is a service provider edge device. The traffic of user networks accesses the MPLS network from CE 1. Priority remarking must be performed for traffic of the three user networks, so that the traffic of different users is processed on the MPLS network according to different priorities. CE 1 must support complex traffic classification and related policies in terms of QoS.

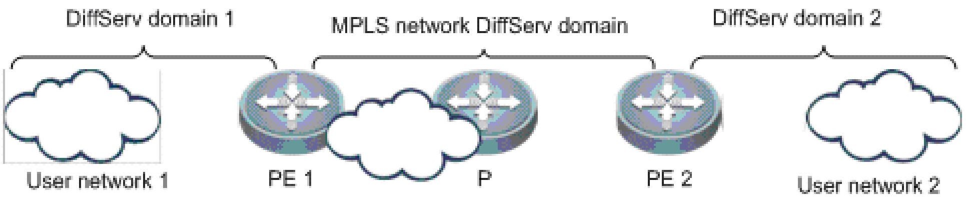
Figure 9 QoS Mapping for User Network Traffic Accessing an MPLS Network



Simple traffic classification is a means to roughly classify packets using simple rules, such as the IP priority or DSCP value of an IP packet, the EXP value of an MPLS packet, or the 802.1p value of a VLAN packet, so as to identify traffic featuring different priorities or Classes of Service (CoSs). In general, simple traffic classification is applied only on core routers in the DiffServ domain.

As shown in Figure 10, PE 1, P, and PE 2 form a backbone network and establish MPLS neighbors with each other. User networks 1 and 2 access the backbone network from the PE. The traffic of user network 1 reaches user network 2 through the backbone network. Service priority mapping must be performed for services between different DiffServ domains.

Figure 10 QoS Mapping for User Network Traffic Across an MPLS Network



Basic Concepts

QoS Priority

Services are classified by QoS requirements into eight types. Packets are first classified and marked after accessing the system. They are treated differently according to packet priorities on the entire forwarding path. The following table defines service priorities.

Table 1 Service Priorities

Code	Service Level		Description
7	CS7		Used for in-band control messages, it represents the highest priority.
6	CS6		Used for protocol packets on the control plane, such as routing protocol packets and Bidirectional Forwarding Detection (BFD) packets.
5	EF (Expedited Forwarding)		Used for services sensitive to delay, jitter, and the packet loss rate, such as VoIP/TDM service packets.
4	AF4	Assured Forwarding	Such services are surely forwarded when they do not exceed the maximum allowed bandwidth. Once the maximum bandwidth is exceeded, however, some packets will be discarded according to a discard priority. Such services are further classified into four types, and different bandwidths are allocated to different service types.
3	AF3		
2	AF2		
1	AF1		
0	BE (Best Effort)		Used for services insensitive to delay, jitter, and the packet loss rate, such as web, FTP, and other Internet services.

QoS Coloring

RFC 2697 and RFC 2698 have defined a service coloring mechanism, which uses three colors Green, Yellow, and Red to implement traffic control for services.

The system colors services based on different service policies to implement different packet discard polices.

Traffic Classifier

In complex traffic classification, it is necessary to define classifiers for various service flows, including classifiers based on different network features such as IPv4, IPv6, MPLS, and VLAN.

In simple traffic classification, traffic classifiers are used to classify traffic based on priorities only, such as DSCP for an IP network, EXP for an MPLS network, and CoS for an 802.1P network.

Traffic Behavior

In complex traffic classification, It is necessary to define the behaviors of various flows, such as Hierarchical QoS (HQoS) marking, priority re-marking, and QoS marking. QoS marking includes priority marking and packet coloring.

In simple traffic classification, however, traffic behaviors support only QoS marking and priority marking.

Traffic Policy

In complex traffic classification, traffic policies are used to associate traffic classifiers with traffic behaviors. One traffic policy can be used to associate multiple traffic classifiers with traffic behaviors so as to perform different operations for different service flows.

In simple traffic classification, a traffic policy consists of an uplink traffic classification mapping table and a downlink traffic classification mapping table. The uplink traffic classification mapping table implements mapping between priorities and QoS levels, whereas the downlink traffic classification mapping table implements mapping from QoS levels to priorities.

⚠ Caution

Simple traffic classification supports mapping between service priorities, CoSs, and discard priorities so as to implement priority bearing and mapping for inter-domain devices.

⚠ Caution

In complex traffic classification, priority marking supports only traffic policies for traffic classification and behaviors on homogeneous networks. For example, the system can perform MPLS priority marking after matching packets with MPLS traffic features.

DiffServ Domain

In the DiffServ model, multiple DiffServ domains are defined and various priority policies are applied. The priority policies for different domains may be different. Therefore, priority mapping must be performed for services across DiffServ domains to guarantee point-to-point QoS.

Universal priority policies are respectively defined for QoS of IP, MPLS, and VLAN networks. DSCP is used for IP networks, EXP for MPLS networks, and CoS for VLAN networks.

Table 2 Default Mapping Between DiffServ Domain DSCP Values and QoS Service Types

DSCP	Service	Color	DSCP	Service	Color
00	BE	Green	32	AF4	Green
01	BE	Green	33	BE	Green
02	BE	Green	34	AF4	Green
03	BE	Green	35	BE	Green
04	BE	Green	36	AF4	Yellow
05	BE	Green	37	BE	Green
06	BE	Green	38	AF4	Red
07	BE	Green	39	BE	Green
08	AF1	Green	40	EF	Green
09	BE	Green	41	BE	Green
10	AF1	Green	42	BE	Green

DSCP	Service	Color	DSCP	Service	Color
11	BE	Green	43	BE	Green
12	AF1	Yellow	44	BE	Green
13	BE	Green	45	BE	Green
14	AF1	Red	46	EF	Green
15	BE	Green	47	BE	Green
16	AF2	Green	48	CS6	Green
17	BE	Green	49	BE	Green
18	AF2	Green	50	BE	Green
19	BE	Green	51	BE	Green
20	AF2	Yellow	52	BE	Green
21	BE	Green	53	BE	Green
22	AF2	Red	54	BE	Green
23	BE	Green	55	BE	Green
24	AF3	Green	56	CS7	Green
25	BE	Green	57	BE	Green
26	AF3	Green	58	BE	Green
27	BE	Green	59	BE	Green
28	AF3	Yellow	60	BE	Green
29	BE	Green	61	BE	Green
30	AF3	Red	62	BE	Green
31	BE	Green	63	BE	Green

Table 3 Default Mapping Between DiffServ Domain QoS Service Types and DSCP Values

Service	Color	DSCP
BE	Green, Yellow, Red	0
AF1	Green	10
AF1	Yellow	12
AF1	Red	14
AF2	Green	18
AF2	Yellow	20
AF2	Red	22
AF3	Green	26
AF3	Yellow	28
AF3	Red	30
AF4	Green	34
AF4	Yellow	36
AF4	Red	38
EF	Green, Yellow, Red	46
CS6	Green, Yellow, Red	48
CS7	Green, Yellow, Red	56

Table 4 Default Mapping Between DiffServ Domain EXP Values and QoS Service Types

EXP	Service	Color
00	BE	Green
01	AF1	Green
02	AF2	Green
03	AF3	Green
04	AF5	Green
05	EF	Green
06	CS6	Green
07	CS7	Green

Table 5 Default Mapping Between DiffServ Domain QoS Service Types and EXP Values

Service	Color	EXP
BE	Green, Yellow, Red	0
AF1	Green, Yellow, Red	1
AF2	Green, Yellow, Red	2
AF3	Green, Yellow, Red	3
AF4	Green, Yellow, Red	4
EF	Green, Yellow, Red	5
CS6	Green, Yellow, Red	6

CS7	Green, Yellow, Red	7

Table 6 Mapping Between CoSs and QoS Service Types

CoS	Service	Color
00	BE	Green
01	BE	Green
02	AF2	Green
03	AF2	Green
04	AF4	Green
05	AF4	Green
06	CS6	Green
07	CS7	Green

Table 7 Default Mapping Between DiffServ Domain QoS Service Types and CoSs

Service	Color	CoS
BE	Green, Yellow, Red	0
AF1	Green, Yellow, Red	1
AF2	Green, Yellow, Red	2
AF3	Green, Yellow, Red	3
AF4	Green, Yellow, Red	4
EF	Green, Yellow, Red	5
CS6	Green, Yellow, Red	6
CS7	Green, Yellow, Red	7

Working Principles

Simple Traffic Classification

Ruijie supports the following simple flow classification:

Simple traffic classification for DiffServ domains of VLAN networks

Simple traffic classification for DiffServ domains of MPLS networks

Simple traffic classification for DiffServ domains of IP networks

Uplink traffic mapping from DiffServ priorities to QoS

Downlink traffic mapping from QoS to DiffServ priorities

Applying simple traffic classification policies based on L3 interfaces or virtual templates

Complex Traffic Classification

Ruijie supports the following complex flow classification:

Complex traffic classification based on L2 information for VLAN networks

Complex traffic classification based on L2 information for MPLS networks

Complex traffic classification based on L2 information for IP networks

Traffic classification policies: associated user queues, priority re-marking, and QoS marking

Priority re-marking across DiffServ domains

Applying simple traffic classification policies based on L3 interfaces or virtual templates

Protocols and Specifications

RFC2474: Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers

RFC2597: Assured Forwarding PHB

RFC2598: Expedited Forwarding PHB

RFC2697: A Single Rate Three Color Marker

RFC2698: A Two Rate Three Color Marker

Default Configurations

The following table shows default traffic classification configurations.

Feature	Default
Complex traffic classification	Disabled
Complex traffic behaviors	None
Complex traffic classification polices	None
Simple traffic classification	Disabled
DiffServ domain in simple traffic classification	A default domain is created
Simple traffic classification policies	The default policy is consistent with the default domain

Configuring Complex Traffic Classification

Configuring Traffic Classifiers

Traffic classifiers are configured to distinguish the traffic of different users from one another so as to provide differentiated services for different users. Each traffic classifier may contain multiple matching rules, and the relationship between the matching rules is determined by the classifier type. If the classifier type is "and", all rules apply to the packet. If the rule type is "or", the packet can match any of the rules. If the classifier type is not specified, the "or" relationship applies between the matching rules. To configure traffic classifiers, perform the following steps:

Command	Purpose
Ruijie(config)# **traffic classifier** *classifier-name* [**and** | **or**]	Enter or create a traffic classifier.
Ruijie(config-traffic-classifier)# **if-match acl** *acl-name* **Or:** Ruijie(config-traffic-classifier)# **if-match dscp** *dscp-value* **Or:** Ruijie(config-traffic-classifier)# **if-match ip-precedence** *ip-precedence-value* **Or:** Ruijie(config-traffic-classifier)# **if-match any**	Set an IPv4 packet matching rule, which can be based on ACL, DSCP, or IP precedence, or any IPv4 packet.
Ruijie(config-traffic-classifier)# **if-match ipv6 acl** *ipv6-acl-number* **Or:** Ruijie(config-traffic-classifier)# **if-match ipv6 dscp** *dscp-value* **Or:** Ruijie(config-traffic-classifier)# **if-match ipv6 any**	Set an IPv6 packet matching rule, which can be based on ACL, DSCP, or any IPv6 packet.
Ruijie(config-traffic-classifier)# **if-match mpls-exp** *mpls-exp-value*	Set an MPLS packet matching rule, which can be based on MPLS EXP.
Ruijie(config-traffic-classifier)# **if-match cos** *cos-value* **Or:** Ruijie(config-traffic-classifier)# **if-match source-mac** *mac-address* **Or:** Ruijie(config-traffic-classifier)# **if-match destination-mac** *mac-address*	Set an Ethernet packet matching rule, which can be based on CoS, source MAC address, or destination MAC address.
Ruijie(config-traffic-classifier)#**exit**	Exit the traffic classifier configuration view.

Configuration example:

# Create a traffic classifier and set an IPv4 packet matching rule based on an ACL:

```
Ruijie(config)#traffic classifier tc1
Ruijie(config-traffic-classifier)#if-match acl 100
Ruijie(config-traffic-classifier)#exit
```

Configuring Traffic Behaviors

Traffic behaviors determine traffic scheduling parameters after traffic is classified. To configure traffic behaviors, perform the following steps:

Command	Purpose
Ruijie(config)# **traffic behavior** *traffic-behavior-name*	Enter or create a traffic behavior.
Ruijie(config-traffic-behavior)#**user-queue** *user-queue-name* [**inbound | outbound**]	Set a user queue.
Ruijie(config-traffic-behavior)#**service-class** *service-class-value* **color** *color-value*	Set the service class and discard priority of packets.
Ruijie(config-traffic-behavior)#**remark dscp** *dscp-value* **Or:** Ruijie(config-traffic-behavior)#**remark ip-precedence** *ip-precedence-value*	Set the remark value of IPv4 packets.
Ruijie(config-traffic-behavior)#**remark ipv6 dscp** *dscp-value*	Set the remark value of IPv6 packets.
Ruijie(config-traffic-behavior)#**remark mpls-exp** *mpls-exp-value*	Set the remark value of MPLS packets.
Ruijie(config-traffic-behavior)#**remark cos** *cos-value*	Set the 802.1Q remark value of Ethernet packets.
Ruijie(config-traffic-behavior)#**sub-traffic-policy** *traffic-policy-name*	Set an associated sub-traffic policy.
Ruijie(config-traffic-behavior)#**rtp-shaping delay** *delay-time* **clock-rate** *clock-rate*	Set RTP shaping behavior.

Configuration example:

# Create a traffic behavior.

```
Ruijie(config)#traffic behavior tb1
Ruijie(config-traffic-behavior)#user-queue uq1 inbound
Ruijie(config-traffic-behavior)#service-class ef color green
Ruijie(config-traffic-behavior)#remark dscp 40
```

Configuring Traffic Policies

Traffic policies associate traffic classifiers with traffic behaviors, so that classified traffic is scheduled according to users' configurations. To configure traffic policies, perform the following steps:

Command	Purpose
Ruijie(config)# **traffic policy** *traffic-policy-name*	Enter or create a traffic policy.
Ruijie(config-traffic-policy)# **classifier** *classifier-name* **behavior** *behavior-name* [**precedence** *precedence-value*]	Specify the traffic behavior for a traffic classifier and set the preference. The smaller the precedence value, the higher the preference.

Configuration example:

# Create a traffic policy.

```
Ruijie(config)#traffic policy tp1
Ruijie(config-traffic-policy)#classifier tc1 behavior tb1 precedence 1
```

Applying Traffic Classification Polices

To apply traffic classification policies, perform the following steps:

Command	Purpose
Ruijie(config)# **interface** *interface-name*	Enter interface configuration mode.

Ruijie(config-if)# **traffic-policy** *traffic-policy-name* **{inbound | outbound} [link-layer | all-layer]**	Apply a traffic policy to the interface. You need to specify the layer parameter. By default, a policy takes effect for L3 and MPLS packets only. If the *link-layer* parameter is specified, the policy takes effect for 802.1P L2 packets only. If the *all-layer* parameter is specified, the policy takes effect for both L3 and L2 packets. If you specify the *link-layer* or *all-layer* parameter, the configured traffic policy is applicable to both the interface and all associated subinterfaces but not merely its subinterfaces. ATM interfaces and subinterfaces do not support the *link-layer* or *all-layer* parameter.

Configuration example:

# Apply a traffic policy to an interface.

```
Ruijie(config)#int gigabitethernet 0/1
Ruijie(config-if-Gigabitethernet 0/1)#traffic-policy tp1 inbound
```

Displaying Configurations

Command	Purpose
show traffic classifier [*classifier-name*]	Show traffic classifier configurations in the system.
show traffic behavior [*behavior-name*]	Show traffic behavior configurations in the system.
show traffic policy [*policy-name*]	Show traffic policy configurations in the system.

Configuration example:

# Show information about the interfaces of the port queue in the system.

```
Ruijie# show traffic classifier tc1
traffic classifier tc1 or
  if-match acl 1501
```

Configuring Simple Traffic Classification

Configuring DiffServ Domains and Traffic Policies

You need to first define DiffServ domains and specify traffic policies for the DiffServ domains during simple traffic classification. To configure DiffServ domains and traffic policies, perform the following steps:

Command	Purpose
Ruijie(config)# **diffserv domain {***ds-domain-name* **| default}**	Enter or create a DiffServ domain.
Ruijie(config-diffserv-domain)# **ip-dscp-inbound** *dscpvalue* **phb** *service-class color* **Or:** Ruijie(config-traffic-classifier)# **ip-dscp-outbound** *serviceclass color* **map** *dscp-value*	Configure an IP traffic policy.
Ruijie(config-diffserv-domain)# **mpls-exp-inbound** *exp* **phb** *service-class color* **Or:** Ruijie(config-diffserv-domain)# **mpls-exp-outbound** *service-class color* **map** *exp-value*	Configure an MPLS traffic policy.
Ruijie(config-diffserv-domain)# **8021p-inbound** *cos-value* **phb** *service-class color* **Or:** Ruijie(config-diffserv-domain)# **8021p-outbound** *service-class color* **map** *cos-value*	Configure an 802.1P traffic policy.
Ruijie(config-traffic-classifier)#**exit**	Exit the traffic classifier configuration view.

Configuration example:

# Create a DiffServ domain named "out-ip" and set the mapping from IP DSCP to QoS.

Ruijie(config)# diffserv domain out-ip

```
Ruijie(config-diffserv-domain)# ip-dscp-inbound 34 phb ef green
Ruijie(config-diffserv-domain)#exit
```

Applying Traffic Classification Polices

To apply traffic classification policies, perform the following steps:

Command	**Purpose**
Ruijie(config)# **interface** *interface-name*	Enter interface configuration mode.
Ruijie(config-if)# **trust upstream** **{** *ds-domain-name* **| default }**	Apply a simple traffic classification policy to the interface.
Ruijie(config-if)# **trust 8021p**	Enable 802.1p simple traffic classification. This configuration is applicable to interfaces only. The configured traffic policy is applicable to both the interface and all associated subinterfaces but not merely its subinterfaces. This command is not available for ATM interfaces or subinterfaces.

Configuration example:

# Apply a traffic policy of the DiffServ domain named "out-ip" to an interface.

```
Ruijie(config)#int gigabitethernet 0/1
Ruijie(config-if-Gigabitethernet 0/1)#trust upstream out-ip
```

Displaying Configurations

Command	**Purpose**
show diffserv domain diffserv-domain-name [8021p-inbound | 8021p-outbound | ip-dscp-inbound | ip-dscp-outbound |mpls-exp-inbound | mpls-exp-outbound]	Show DiffServ domain configurations.

Configuration example:

# Show configuration information about the DiffServ domain named "ip-out".

```
Ruijie# show diffserv domain ipdscp
IP-DSCP map to Server-class and Color :
  0 --> be   green
  1 --> be   green
  2 --> be   green
  3 --> be   green
  4 --> be   green
  5 --> be   green
  6 --> be   green
  7 --> be   green
  8 --> af1  green
  9 --> be   green
 10 --> af1  green
 11 --> be   green
 12 --> af1  yellow
 13 --> be   green
 14 --> af1  red
 15 --> be   green
 16 --> af2  green
 17 --> be   green
 18 --> af2  green
 19 --> be   green
 20 --> af2  yellow
 21 --> be   green
 22 --> af2  red
 23 --> be   green
 24 --> af3  green
 25 --> be   green
 26 --> af3  green
 27 --> be   green
 28 --> af3  yellow
 29 --> be   green
 30 --> af3  red
 31 --> be   green
 32 --> af4   green
 33 --> be    green
 34 --> af4   green
 35 --> be    green
 36 --> af4   yellow
 37 --> be    green
 38 --> af4   red
 39 --> be    green
 40 --> ef    green
 41 --> be    green
 42 --> be    green
 43 --> be    green
 44 --> be    green
 45 --> be    green
 46 --> ef    green
 47 --> be    green
 48 --> cs6   green
 49 --> be    green
 50 --> be    green
 51 --> be    green
 52 --> be    green
 53 --> be    green
 54 --> be    green
 55 --> be    green
 56 --> cs7   green
 57 --> be    green
 58 --> be    green
 59 --> be    green
 60 --> be    green
 61 --> be    green
 62 --> be    green
 63 --> be    green

MPLS-EXP map to Server-class and Color :
  0 --> be    green
  1 --> af1   green
  2 --> af2   green
  3 --> af3   green
  4 --> af4   green
  5 --> ef    green
  6 --> cs6   green
  7 --> cs7   green

VLAN-Cos map to Server-class and Color :
  0 --> be    green
  1 --> af1   green
  2 --> af2   green
  3 --> af3   green
  4 --> af4   green
  5 --> ef    green
  6 --> cs6   green
  7 --> cs7   green

Server-class and Color map to IP-DSCP :
 be    green    -->  0
 be    yellow   -->  0
 be    red      -->  0
 af1   green    --> 10
 af1   yellow   --> 12
 af1   red      --> 14
 af2   green    --> 18
 af2   yellow   --> 20
 af2   red      --> 22
 af3   green    --> 26
 af3   yellow   --> 28
 af3   red      --> 30
 af4   green    --> 34
 af4   yellow   --> 36
 af4   red      --> 38
 ef    green    --> 46
 ef   yellow  --> 46
 ef   red     --> 46
 cs6  green   --> 48
 cs6  yellow  --> 48
 cs6  red     --> 48
 cs7  green   --> 56
 cs7  yellow  --> 56
 cs7  red     --> 56

Server-class and Color map to MPLS-EXP :
 be   green   -->  0
 be   yellow  -->  0
 be   red     -->  0
 af1  green   -->  1
 af1  yellow  -->  1
 af1  red     -->  1
 af2  green   -->  2
 af2  yellow  -->  2
 af2  red     -->  2
 af3  green   -->  3
 af3  yellow  -->  3
 af3  red     -->  3
 af4  green   -->  4
 af4  yellow  -->  4
 af4  red     -->  4
 ef   green   -->  5
 ef   yellow  -->  5
 ef   red     -->  5
 cs6  green   -->  6
 cs6  yellow  -->  6
 cs6  red     -->  6
 cs7  green   -->  7
 cs7  yellow  -->  7
 cs7  red     -->  7

Server-class and Color map to VLAN-CoS :
 be   green   -->  0
 be   yellow  -->  0
 be   red     -->  0
 af1  green   -->  1
 af1  yellow  -->  1
 af1  red     -->  1
 af2  green   -->  2
 af2  yellow  -->  2
 af2  red     -->  2
 af3  green   -->  3
 af3  yellow  -->  3
 af3  red     -->  3
 af4  green   -->  4
 af4  yellow  -->  4
 af4  red     -->  4
 ef   green   -->  5
 ef   yellow  -->  5
 ef   red     -->  5
 cs6  green   -->  6
 cs6  yellow  -->  6
 cs6  red     -->  6
 cs7  green   -->  7
 cs7  yellow  -->  7
 cs7  red     -->  7
```

Examples for Configuring Traffic Classification

Configuration Example 1

Networking Requirements

- Device requirements:

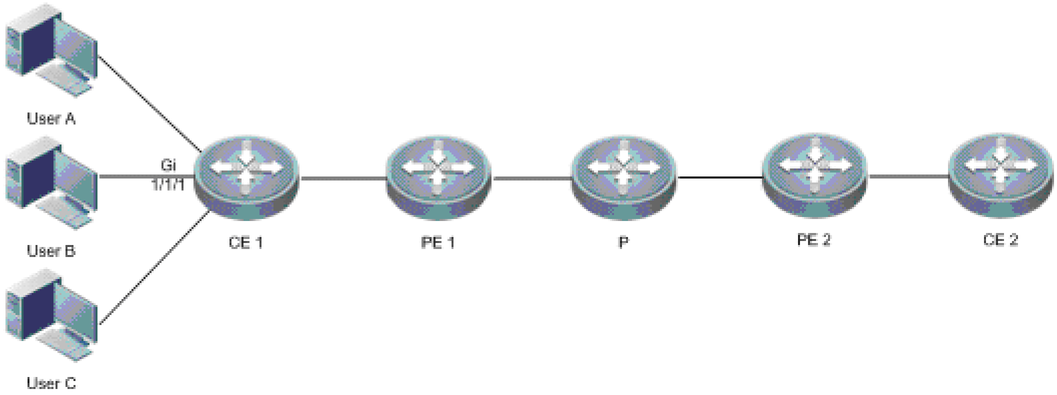
Four routers.

- Configuration requirements:

Two local users access an MPLS backbone network from a CE. The committed access rate of user A is 60 Mbps, and the peak access rate is 80 Mbps. The committed access rate of user B is 30 Mbps, and the peak access rate is 40 Mbps. The IP address of user A is 10.1.10.1, and that of user B is 10.1.10.2.

Network Topology

Figure 11 Network Topology for Complex Traffic Classification



Configuration Tips

Configure traffic classifiers to identify users.

Configure traffic behaviors to monitor and control user traffic.

Configuration Steps

Configure traffic classifiers:

```
# Configure a traffic classifier named "tc1".
Ruijie(config)#access-list 100 permit ip host 10.1.10.1 any
Ruijie(config)#traffic classifier tc1
Ruijie(config-traffic-classifier)#if-match acl 100
Ruijie(config-traffic-classifier)#exit
# Configure a traffic classifier named "tc2".
Ruijie(config)# access-list 110 permit ip host 10.1.10.2 any
Ruijie(config)#traffic classifier tc2
Ruijie(config-traffic-classifier)#if-match acl 110
Ruijie(config-traffic-classifier)#exit
```

Configure user queues:

```
# Configure a user queue named "uq1".
Ruijie(config)#user-queue uq1 inbound
Ruijie(config-user-queue)#cir 60000 pir 80000
Ruijie(config-user-queue)#exit
```

```
# Configure a user queue named "uq2".
Ruijie(config)#user-queue uq2 inbound
Ruijie(config-user-queue)#cir 30000 pir 40000
Ruijie(config-user-queue)#exit
```

Configure traffic behaviors:

```
#Configure a traffic behavior named "tb1".
Ruijie(config)#traffic behavior tb1
Ruijie(config-traffic-behavior)#user-queue uq1 inbound
Ruijie(config-traffic-behavior)#exit
```

```
#Configure a traffic behavior named "tb2".
Ruijie(config)#traffic behavior tb2
Ruijie(config-traffic-behavior)#user-queue uq2 inbound
Ruijie(config-traffic-behavior)#exit
```

Configure a traffic policy:

```
Ruijie(config)#traffic policy tp1
Ruijie(config-traffic-policy)#classifier tc1 behavior tb1
Ruijie(config-traffic-policy)#classifier tc2 behavior tb2
Ruijie(config-traffic-policy)#exit
```

Apply the traffic policy to an interface:

```
Ruijie(config)#int gigabitethernet 1/1/1
Ruijie(config-if-Gigabitethernet1/1/1)#traffic-policy tp1 inbound
```

Verification

Show traffic policy information:

```
Ruijie# show traffic policy tp1
traffic policy tp1
  classifier tc1 behavior tb1 precedence 1
  classifier tc2 behavior tb2 precedence 2
```

Configuration Example 2

Networking Requirements

- Device requirements:

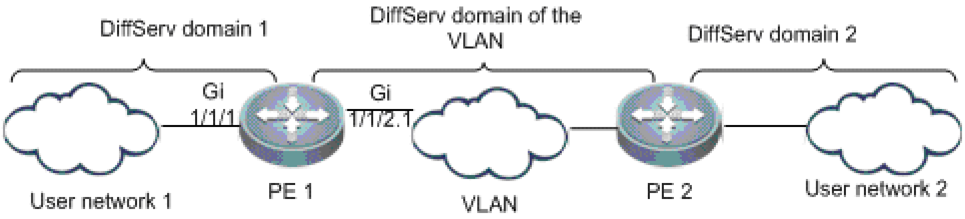
One Ethernet switch and two routers.

- Configuration requirements:

Two routers provide access services for the users of user network 1 and user network 2 respectively. The service priority of users of user network 1 must be able to be carried and transferred to user network 2 to implement DiffServ domain mapping across the VLAN. It is necessary to add HQOS configurations on PE 1.

Network Topology

Figure 12 User Network Interconnection Across a VLAN



Configuration Tips

Configure DiffServ domains on routers.

Apply simple traffic classification to the user networks and VLAN.

Configuration Steps

Create DiffServ domain policies:

```
# Configure a traffic classifier named "usera".
Ruijie(config)# diffserv domain usera
Ruijie(config-diffserv-domain)#ip-dscp-inbound 34 phb ef green
Ruijie(config-diffserv-domain)#ip-dscp-inbound 16 phb be green
# Configure a traffic classifier named "vlan".
Ruijie(config)# diffserv domain vlan
Ruijie(config-diffserv-domain)# 8021p-outbound ef green map 4
Ruijie(config-diffserv-domain)# 8021p-outbound be green map 1
```

Apply simple traffic classifiers to respective interfaces:

```
Ruijie(config)#int gigabitethernet 1/1/1
Ruijie(config-if-Gigabitethernet1/1/1)#trust upstream vlan
Ruijie(config-if-Gigabitethernet1/1/1)#exit

Ruijie(config)#int gigabitethernet 1/1/2.1
Ruijie(config-if-Gigabitethernet1/1/2.1)#trust upstream usera
Ruijie(config-if-Gigabitethernet1/1/2.1)#trust 8021p
Ruijie(config-if-Gigabitethernet1/1/2.1)#exit
```

Verification

Show DiffServ domain configurations:

```
Ruijie# show diffserv domain usera
IP-DSCP map to Server-class and Color:
  0 --> be   green
  1 --> be   green
  2 --> be   green
  3 --> be   green
  4 --> be   green
  5 --> be   green
  6 --> be   green
  7 --> be   green
  8 --> af1  green
  9 --> be   green
 10 --> af1  green
 11 --> be   green
 12 --> af1  yellow
 13 --> be   green
 14 --> af1  red
 15 --> be   green
 16 --> be   green
 17 --> be   green
 18 --> af2  green
 19 --> be   green
 20 --> af2  yellow
 21 --> be   green
 22 --> af2  red
 23 --> be   green
 24 --> af3  green
 25 --> be   green
 26 --> af3  green
 27 --> be   green
 28 --> af3  yellow
 29 --> be   green
 30 --> af3  red
 31 --> be   green
 32 --> af4  green
 33 --> be   green
 34 --> ef   green
 35 --> be   green
 36 --> af4  yellow
 37 --> be   green
 38 --> af4  red
 39 --> be   green
 40 --> ef   green
 41 --> be   green
 42 --> be   green
 43 --> be   green
 44 --> be   green
 45 --> be   green
 46 --> ef   green
 47 --> be   green
 48 --> cs6  green
 49 --> be   green
 50 --> be   green
 51 --> be   green
 52 --> be   green
 53 --> be   green
 54 --> be   green
 55 --> be   green
 56 --> cs7  green
 57 --> be   green
 58 --> be   green
 59 --> be   green
 60 --> be   green
 61 --> be   green
 62 --> be   green
 63 --> be   green
```

```
Ruijie# show diffserv domain vlan
Server-class and Color map to VLAN-CoS :
 be    green    -->  1
 be    yellow   -->  0
 be    red      -->  0
 af1   green    -->  1
 af1   yellow   -->  1
 af1   red      -->  1
 af2   green    -->  2
 af2   yellow   -->  2
 af2   red      -->  2
 af3   green    -->  3
 af3   yellow   -->  3
 af3   red      -->  3
 af4   green    -->  4
 af4   yellow   -->  4
 af4   red      -->  4
 ef    green    -->  4
 ef    yellow   -->  5
 ef    red      -->  5
 cs6   green    -->  6
 cs6   yellow   -->  6
 cs6   red      -->  6
 cs7   green    -->  7
 cs7   yellow   -->  7
 cs7   red      -->  7
```

Configuration Example 3

Networking Requirements

- Device requirements:

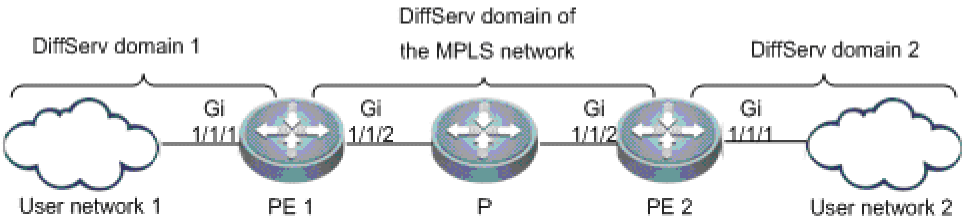
Five routers.

- Configuration requirements:

User network 1 and user network 2 access an MPLS backbone network from PE 1 and PE 2 respectively. The service priority of users of user network 1 must be able to be carried and transferred to user network 2 to implement DiffServ domain mapping across the MPLS network. It is necessary to add HQOS configurations on PE 1 and PE 2.

Network Topology

Figure 13 User Network Interconnection Across an MPLS Network



Configuration Tips

Configure DiffServ domains on routers.

Apply simple traffic classification to the user networks and MPLS network.

Configuration Steps

Configure PE 1 and create DiffServ domain policies:

```
# Configure a traffic classifier named "usera".
Ruijie(config)# diffserv domain usera
Ruijie(config-diffserv-domain)#ip-dscp-inbound 34 phb ef green
Ruijie(config-diffserv-domain)#ip-dscp-inbound 16 phb be green
# Configure a traffic classifier named "mpls".
Ruijie(config)# diffserv domain mplsa
Ruijie(config-diffserv-domain)# mpls-exp-outbound ef green map 4
Ruijie(config-diffserv-domain)# mpls-exp-outbound be green map 1
```

Configure PE1. Apply simple traffic classifiers to respective interfaces:

```
Ruijie(config)#int gigabitethernet 1/1/1
Ruijie(config-if-Gigabitethernet1/1/1)#trust upstream usera
Ruijie(config-if-Gigabitethernet1/1/1)#exit

Ruijie(config)#int gigabitethernet 1/1/2
Ruijie(config-if-Gigabitethernet1/1/2)#trust upstream mplsa
Ruijie(config-if-Gigabitethernet1/1/2)#exit
```

Configure PE 2 and create DiffServ domain policies:

```
# Configure a traffic classifier named "usera".
Ruijie(config)# diffserv domain userb
Ruijie(config-diffserv-domain)# ip-dscp-outbound ef green map
34Ruijie(config-diffserv-domain)# ip-dscp-outbound be green map 16
# Configure a traffic classifier named "mpls".
Ruijie(config)# diffserv domain mplsb
Ruijie(config-diffserv-domain)# mpls-exp-inbound 4 phb ef green
Ruijie(config-diffserv-domain)# mpls-exp-inbound 1 phb be green
```

Configure PE2. Apply simple traffic classifiers to respective interfaces:

```
Ruijie(config)#int gigabitethernet 1/1/1
Ruijie(config-if-Gigabitethernet1/1/1)#trust upstream userb
Ruijie(config-if-Gigabitethernet1/1/1)#exit

Ruijie(config)#int gigabitethernet 1/1/2
Ruijie(config-if-Gigabitethernet1/1/2)#trust upstream mplsb
Ruijie(config-if-Gigabitethernet1/1/2)#exit
```

Verification

Show DiffServ domain configurations on PE 1:

```
Ruijie# show diffserv domain usera
IP-DSCP map to Server-class and Color :
  0 --> be   green
  1 --> be   green
  2 --> be   green
  3 --> be   green
  4 --> be   green
  5 --> be   green
  6 --> be   green
  7 --> be   green
  8 --> af1  green
  9 --> be   green
 10 --> af1  green
 11 --> be   green
 12 --> af1  yellow
 13 --> be   green
 14 --> af1  red
 15 --> be   green
 16 --> be   green
 17 --> be   green
 18 --> af2  green
 19 --> be   green
 20 --> af2  yellow
 21 --> be   green
 22 --> af2  red
 23 --> be   green
 24 --> af3  green
 25 --> be   green
 26 --> af3  green
 27 --> be   green
 28 --> af3  yellow
 29 --> be   green
 30 --> af3  red
 31 --> be   green
 32 --> af4  green
 33 --> be   green
 34 --> ef   green
 35 --> be   green
 36 --> af4  yellow
 37 --> be   green
 38 --> af4  red
 39 --> be   green
 40 --> ef   green
 41 --> be   green
 42 --> be   green
 43 --> be   green
 44 --> be   green
 45 --> be   green
 46 --> ef   green
 47 --> be   green
 48 --> cs6  green
 49 --> be   green
 50 --> be   green
 51 --> be   green
 52 --> be   green
 53 --> be   green
 54 --> be   green
 55 --> be   green
 56 --> cs7  green
 57 --> be   green
 58 --> be   green
 59 --> be   green
 60 --> be   green
 61 --> be   green
 62 --> be   green
 63 --> be   green
Ruijie# show diffserv domain mplsa
Server-class and Color map to MPLS-EXP :
 be   green   -->  1
 be   yellow  -->  0
 be   red     -->  0
 af1  green   -->  1
 af1  yellow  -->  1
 af1  red     -->  1
 af2  green   -->  2
 af2  yellow  -->  2
 af2  red     -->  2
 af3  green   -->  3
 af3  yellow  -->  3
 af3  red     -->  3
 af4  green   -->  4
 af4  yellow  -->  4
 af4  red     -->  4
 ef   green   -->  4
 ef   yellow  -->  5
 ef   red     -->  5
 cs6  green   -->  6
 cs6  yellow  -->  6
 cs6  red     -->  6
 cs7  green   -->  7
 cs7  yellow  -->  7
 cs7  red     -->  7
```

Show DiffServ domain configurations on PE 2:

```
Ruijie# show diffserv domain userb
Server-class and Color map to IP-DSCP :
 be    green    --> 16
 be    yellow   -->  0
 be    red      -->  0
 af1   green    --> 10
 af1   yellow   --> 12
 af1   red      --> 14
 af2   green    --> 18
 af2   yellow   --> 20
 af2   red      --> 22
 af3   green    --> 26
 af3   yellow   --> 28
 af3   red      --> 30
 af4   green    --> 34
 af4   yellow   --> 36
 af4   red      --> 38
 ef    green    --> 34
 ef    yellow   --> 46
 ef    red      --> 46
 cs6   green    --> 48
 cs6   yellow   --> 48
 cs6   red      --> 48
 cs7   green    --> 56
 cs7   yellow   --> 56
 cs7   red      --> 56
Ruijie# show diffserv domain mplsb
MPLS-EXP map to Server-class and Color :
  0 --> be    green
  1 --> be    green
  2 --> af2   green
  3 --> af3   green
  4 --> ef    green
  5 --> ef    green
  6 --> cs6   green
  7 --> cs7   green
```

Configuring HQoS

Understanding HQoS

HQoS Overview

The traditional QoS offers different treatment to services and ensure QoS needs such as bandwidth and delay by classifying service flows and specifying policies for them. However, the existing user access network is complex, and there are a large number of access devices (such as Layer 2 switches and converters) that do not support complex QoS. Although egress devices of the user access network can ensure QoS for transmitted services as much as possible, they cannot ensure QoS based on the user and user group in a more refined manner.

Hierarchical QoS (HQoS) is different from the traditional QoS, which is specific to services. HQoS can provide QoS for data flows in the network by service, user, floor, and residential area. Service QoS provides quality service for multiple service flows of a single user. User QoS provides quality service for multiple users on a floor, and so on for floor QoS and residential area QoS. HQoS can implement more refined service assurance for data aggregation devices to improve service quality for whole-network users.

Basic Concepts

FQ

The flow queue (FQ) indicates service queues of a user. Each user has eight FQs, which correspond to eight service priorities respectively (BE, AF1, AF2, AF3, AF4, EF, CS6, and CS7). The eight FQs can be configured with Priority Queuing (PQ), Weighted Fair Queuing (WFQ), or Low-priority Queuing (LPQ). Each FQ supports Weighted Random Early Detection (WRED) and traffic shaping.

UQ

The user here indicates one VLAN or VPN. Users can be divided based on the interface, sub-interface, or ACL. Each user has a user queue (UQ), which is aggregated by eight fixed FQs. You can limit the total bandwidth of each user by performing rate limiting to its UQ.

Note

If users belong to different line cards of a distributed device, the UQ function takes effect based on each line card.

GQ

Multiple users can be bound to one user group. Each user group has a group queue (GQ). You can control the traffic of a user group by performing traffic shaping to the GQ.

Note If user groups belong to different line cards of a distributed device, the GQ function takes effect based on each line card.

Destination Interface/Destination Device CQ

Each destination interface has eight queues, which correspond to eight service types respectively. The eight FQs can be configured with SP and WFQ. Each class queue (CQ) supports WRED and traffic shaping.

The destination device CQ can be regarded as an uplink destination interface CQ. They are the same in nature. The difference is that the packets of the destination device CQ are sent to boards instead of interfaces. Each destination device correspond to four queues, which correspond to four service types. The four queues are scheduled in SP mode.

LPQ

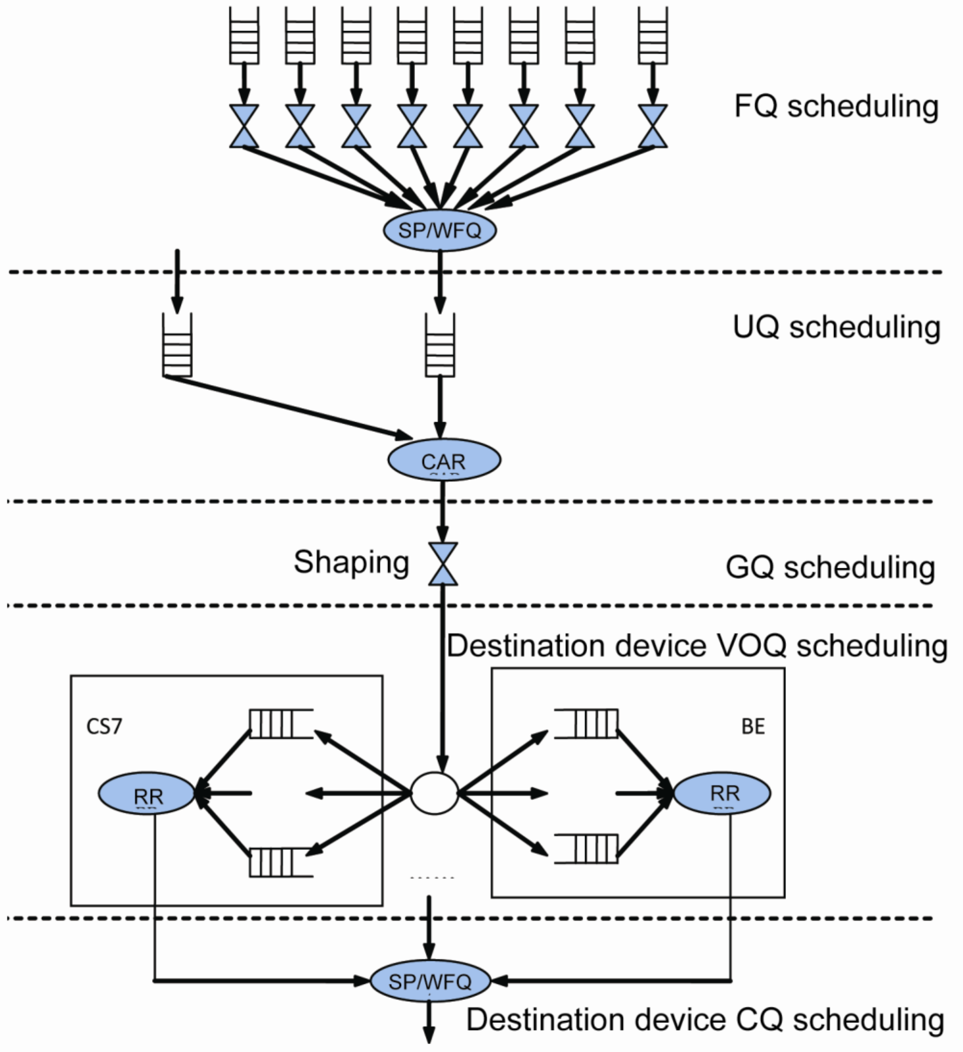
There are three scheduling mechanisms in HQoS: PQ, WFQ, and Low-priority Queue (LPQ), which descend in order of priority. In the case of congestion, PQ queues and WFQ queues can preempt the bandwidth of LPQ queues.

How HQoS Works

Uplink HQoS Scheduling

Uplink HQoS scheduling is divided into five levels: CQ > UQ > GQ > Destination device VOQ > Destination device CQ, as shown in the following figure:

Figure 1 Uplink HQoS scheduling process



Level 1 scheduling is FQ scheduling. The FQ is an entity queue, that is, the packets are cached in an FQ. FQ includes FQ WRED, FQ shaping, and FQ scheduling. WERD congestion avoidance is required before a packet enters a FQ. The system supports discarding priorities of three colors: red, yellow, and green. Red indicates the highest discarding priority, and green indicates the lowest one. Each FQ sets the minimum threshold, maximum threshold, and discarding probability for each discarding priority. The higher the discarding priority, the greater the maximum threshold and minimum threshold, and the greater the discarding probability of packets. Traffic shaping is performed via token bucket after WRED for FQ. Finally, FQ scheduling is performed by using Strict Priority (SP)+WFQ. SP includes PQ and LPQ. CS7, CS6, EF, AF, and BE are scheduled in sequence. Low-priority queues are scheduled only when there are no packets in high-priority queues. AF includes four types, which are scheduled according to their respective weights.

Level 2 scheduling is UQ scheduling. The UQ is a virtual queue, that is, the packets are not cached in a UQ. UQ scheduling is only an HQoS scheduling level. Each UQ includes eight FQs that share the bandwidth of the UQ. The UQ supports committed interface rate (CIR) only. Each UQ can invoke zero to one GQ.

Level 3 scheduling is GQ scheduling. The GQ is a virtual queue. Multiple UQs can be bound to a GQ for scheduling. The GQ supports traffic shaping only. If no UQ is bound to a GQ, GQ scheduling will be skipped.

Level 4 scheduling is VoQ scheduling. VoQ scheduling is the uplink traffic scheduling among line cards. The VoQ breaks into four groups by service priority, with one queue for each destination device in each group. When packets pass through UQ scheduling, they enter different VoQs based on their destination devices and priorities. VoQ scheduling is configured by the system, and no additional user configuration is required.

Level 5 scheduling is CQ scheduling. The CQ is a virtual queue, instead of an entity queue that caches packets. Each CQ, however, maintains information such as scheduling priority and weigh, and supports SP/WFQ scheduling. CQ scheduling for uplink HQoS is configured by the system, and no additional user configuration is required.

- CoS-based uplink HQoS processing:

Flows are classified based on configured flow classification rules, and are prioritized at eight levels.

The UQ, GQ, WRED parameters, and scheduling policies of packets are determined based on flow behavior rules. Then, packets are passed to corresponding FQs.

- FQ scheduling can be performed according to user configuration to avoid congestion.
- An FQ can be scheduled as a PQ, WFQ, or LPQ. The PQ and WFQ can preempt the bandwidth of the LPQ.

The remaining bandwidth for the GQ is checked. If the remaining bandwidth for the GQ is insufficient, the UQ included in the GQ is not scheduled. Otherwise, the UQ included in the GQ is scheduled. The bandwidth for the GQ is configurable.

The remaining bandwidth for the UQ is checked. If the remaining bandwidth for the UQ is insufficient, the FQ included in the UQ is not scheduled. Otherwise, the FQ is scheduled. The bandwidth for the UQ is configurable.

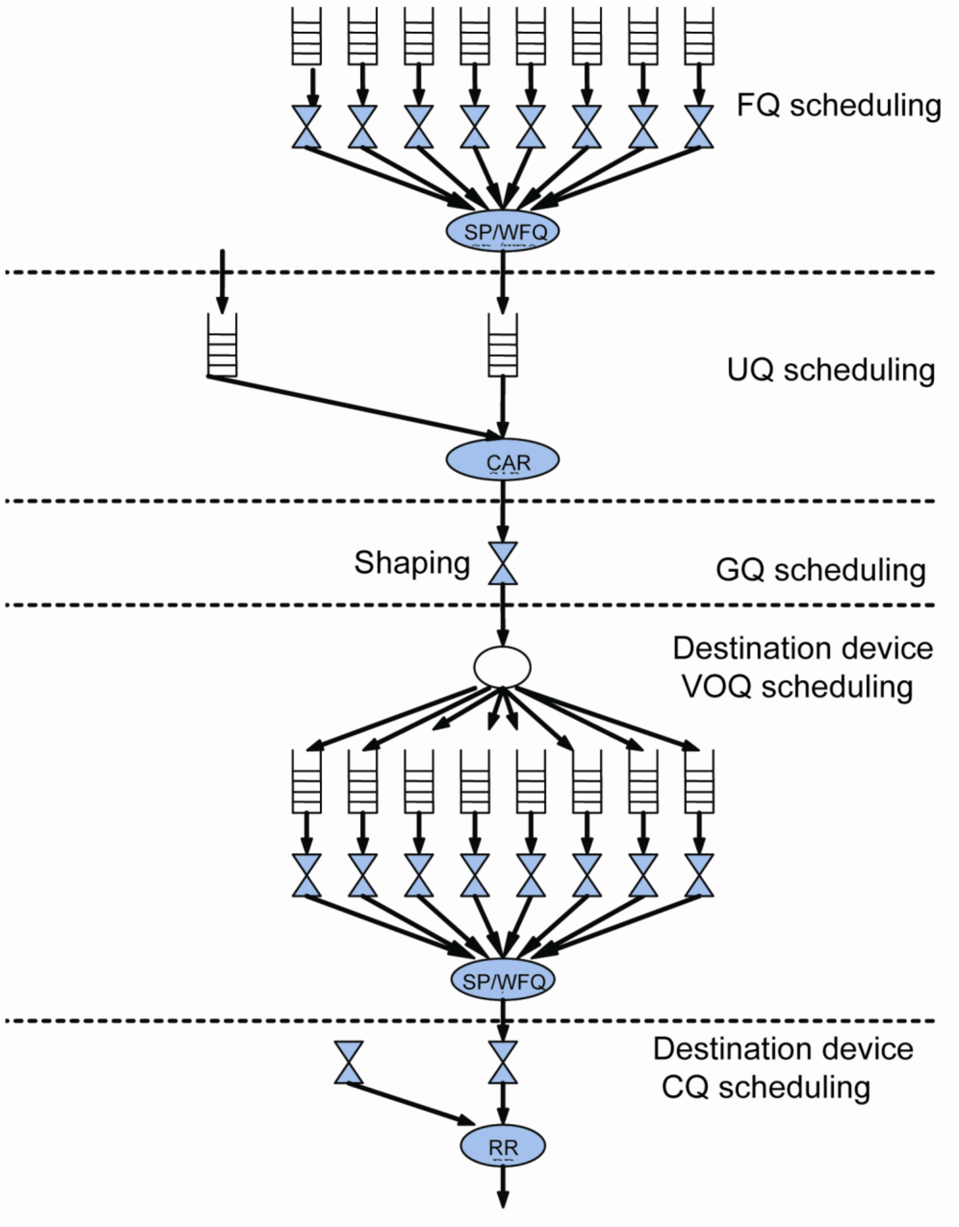
After scheduling, packets in the FQ enter the VoQs on destination devices or destination interfaces.

CQ scheduling is performed in SP mode among four queues. After scheduling, packets are forwarded.

Downlink HQoS Scheduling

Downlink HQoS scheduling is divided into five levels: FQ > UQ > GQ > Interface CQ > Destination interface, as shown in the following figure:

Figure 2 Downlink HQoS Scheduling Process



Level 1 FQ scheduling, level 2 UQ scheduling, and level 3 GQ scheduling are identical with those of uplink HQoS scheduling.

Level 4 scheduling is CQ scheduling. There are eight CQs on each interface, corresponding to eight service priorities. Different from the case of uplink HQoS scheduling, you can configure CQ congestion avoidance parameters and the traffic shaping value. The CQ is scheduled in SP + WFQ mode.

Level 5 scheduling is destination interface scheduling. The system polls interfaces for scheduling. Level 5 scheduling is configured by the system, and no additional user configuration is required.

Protocol Specifications

TR-059: A layered QoS model defined by the DSL Forum

Default Configuration

The following table describes the default HQoS configurations:

Function	Default
WRED queue lower threshold	20
WRED queue upper threshold	40
WRED queue discarding probability	100

Configuring CoS-based HqoS

Configuring a Traffic Classifier Rule

Configuring a traffic classifier rule is to distinguish between user flows to provide differentiated services for different users. Each traffic classifier rule can contain multiple match rules, the relationship among which is determined by the rule type. If the rule type is "and", the packet must match all rules; if the rule type is "or", the packet can match any one of the rules. If no rule type is specified, "or" rule type is adopted by default. Configure this function as follows:

Command	Purpose
Ruijie(config)# **traffic classifier** *classifier-name* [**and** | **or**]	Enters/Creates a traffic classifier rule.
Ruijie(config-traffic-classifier)# **if-match acl** *acl-number* Or: Ruijie(config-traffic-classifier)# **if-match dscp** *dscp-value* Or: Ruijie(config-traffic-classifier)# **if-match ip-precedence** *ip-precedence-value* Or: Ruijie(config-traffic-classifier)# **if-match any**	Sets IPv4 packet match rules. Supports matching packets based on ACL, DSCP, and IP precedence and matching all IPv4 packets (any).
Ruijie(config-traffic-classifier)# **if-match ipv6** *ipv6-acl-number* Or: Ruijie(config-traffic-classifier)# **if-match ipv6 dscp** *dscp-value* Or: Ruijie(config-traffic-classifier)# **if-match ipv6 any**	Sets IPv6 packet match rules. Supports matching packets based on ACL and DSCP, and matching all IPv6 packets (any).
Ruijie(config-traffic-classifier)# **if-match mpls-exp** *mpls-exp-value*	Sets MPLS packet match rules. Supports matching packets based on MPLS EXP.
Ruijie(config-traffic-classifier)# **if-match cos** *cos-value* Or: Ruijie(config-traffic-classifier)# **if-match source-mac** *mac-address* Or: Ruijie(config-traffic-classifier)# **if-match destination-mac** *mac-address*	Sets Ethernet packet match rules. Supports matching packets based on CoS, source MAC address, and destination MAC address.
Ruijie(config-traffic-classifier)#**exit**	Exits the traffic classifier rule configuration.

Configuration example:

# Create a traffic classifier rule and set the IPv4 packet match rule to be based on ACL.

```
Ruijie(config)#traffic classifier tc1
Ruijie(config-traffic-classifier)#if-match acl 100
Ruijie(config-traffic-classifier)#exit
```

Configuring an FQ WRED Template

Configuring a WRED template is to configure congestion avoidance parameters, including upper threshold, lower threshold, and discarding probability, for packets in three colors. When the number of packets in a queue is smaller than the lower threshold, packets are not discarded. When the number of packets in a queue is larger than the lower threshold and smaller than the upper threshold, packets are discarded at certain probability. When the number of packets in a queue is larger than the upper threshold, packets are discarded. You can configure different WRED templates and apply them to different FQs.

Configure this function as follows:

Command	Purpose
Ruijie(config)# **wred** *wred-template-name*	Creates/Enters a WRED template.

Ruijie(config-wred)#**color green low-limit** *low-limit-threshold* **high-limit** *high-limit-threshold* **discard-percent** *discard-percent-value* Ruijie(config-wred)#**color yellow low-limit** *low-limit-threshold* **high-limit** *high-limit-threshold* **discard-percent** *discard-percent-value* Ruijie(config-wred)#**color red low-limit** *low-limit-threshold* **high-limit** *high-limit-threshold* **discard-percent** *discard-percent-value*	Sets the upper threshold, lower threshold, and probability for queues in three colors.
Ruijie(config- wred)#**exit**	Exits the WRED template configuration.

Configuration example:

# Create a WRED template and set congestion avoidance parameters.

```
Ruijie(config)#wred wt1
Ruijie(config-wred)#color green low-limit 40 high-limit 60 discard-percent 10
Ruijie(config-wred)#color yellow low-limit 30 high-limit 50 discard-percent 10
Ruijie(config-wred)#color red low-limit 20 high-limit 40 discard-percent 10
Ruijie(config-wred)#exit
```

Configuring an FQ Template

Configuring an FQ template is to configure the scheduling mode (PQ, WFQ, and LPQ), traffic shaping value, and WRED parameters of eight kinds of PQs. PQ scheduling is performed on queues with the priority as ef, cs6, or cs7. WFQ scheduling is performed on queues with the priority as be, af1, af2, af3, or af4. Traffic shaping is not performed by default and the default discarding policy is tail discarding. You can configure multiple FQ templates and apply them to different FQs.

Configure this function as follows:

Command	**Purpose**
Ruijie(config)# **flow-queue** *flow-queue-template-name*	Enters/Creates an FQ template.
Ruijie(config-flow-queue)# **queue** *cos-value* {**pq** | **wfq weight** *weight-value* | **lpq**} [**shaping** *shaping-value*] [**wred** *wred-name*] [**depth** *depth-value*]	Set FQ scheduling parameters.
Ruijie(config- flow-queue)#**exit**	Exits the FQ template configuration.

Configuration example:

# Create an FQ template and set FQ scheduling parameters.

```
Ruijie(config)#flow-queue fqt1
Ruijie(config-flow-queue)# queue be lpq
Ruijie(config-flow-queue)# queue af1 wfq weight 10 shaping 100000 wred wt1
Ruijie(config-flow-queue)# queue cs7 pq shaping wred wt1
Ruijie(config-flow-queue)#exit
```

Configuring an FQ Mapping Template

Configuring an FQ mapping template is to configure the mapping from eight kinds of PQs to CQs.

Configure this function as follows:

Command	**Purpose**
Ruijie(config)# **flow-mapping** *flow-mapping name*	Enters/Creates an FQ mapping template.
Ruijie(config-flow-mapping)# **map flow-queue** *cos-value* **to port-queue** *cos-value*	Sets the mapping from FQ to CQ.
Ruijie(config- flow-mapping)#**exit**	Exits the FQ mapping template configuration.

Configuration example:

# Create a FQ mapping template and set the mapping from FQ to CQ.

```
Ruijie(config)#flow-mapping fmt1
Ruijie(config-flow-mapping)# map flow-queue af1 to port-queue ef
Ruijie(config-flow-mapping)#exit
```

Configuring a UQ

Configuring a UQ consists of configuring the CIR, configuring the FQ template, and associating a GQ.

Configure this function as follows:

Command	Purpose
Ruijie(config)# **user-queue** *user-queue-name* **inbound | outbound**	Enters/Creates a UQ.
Ruijie(config-user-queue)# **cir** *vir-value* **pir** *pir-value*	Sets UQ traffic shaping parameters. By default, the CIR of the UQ is 0.
Ruijie(config-user-queue)# **flow-queue** *flow-queue-template-name*	Sets UQ FQ scheduling parameters.
Ruijie(config-user-queue)# **user-group-queue** *user-group-queue-name*	Associates the UQ with a GQ.
Ruijie(config-user-queue)# **flow-mapping** *flow-rmap-name*	Sets an FQ mapping template.

Configuration example:

# Create a UQ and set scheduling parameters.

```
Ruijie(config)#user-queue uq1 inbound
Ruijie(confg-user-queue)#cir 100000 pir 100000
Ruijie(config-user-queue)#flow-queue fqt1
Ruijie(config-user-queue)#user-group-queue ugq1
Ruijie(config-user-queue)#flow-mapping fmt1
```

Configuring a GQ

A GQ is a bundle of multiple UQs for centralized traffic shaping.

Configure this function as follows:

Command	Purpose
Ruijie(config)# **user-group-queue** *user-group-queue-name* [**inbound | outbound**]	Enters/Creates a GQ.
Ruijie(config-user-group-queue)# **shaping** *shaping-value*	Sets the traffic shaping value of the GQ.

Configuration example:

# Create a GQ and set the traffic shaping value.

```
Ruijie(config)#user-group-queue ugq1 inbound
Ruijie(config-user-group-queue)#shaping 100000
```

Configuring a Traffic Behavior Rule

The traffic behavior rule determines traffic scheduling parameters after classification. Configure this function as follows:

Command	Purpose
Ruijie(config)# **traffic behavior** *traffic-behavior-name*	Enters/Creates a traffic behavior rule.
Ruijie(config-traffic-behavior)#**user-queue** *user-queue-name* [**inbound | outbound**]	Sets the UQ.
Ruijie(config-traffic-behavior)#**service-class** *service-class-value* **color** *color-value*	Sets the color of packets with different priorities.
Ruijie(config-traffic-behavior)#**remark dscp** *dscp-value* **Or:** Ruijie(config-traffic-behavior)#**remark ip-precedence** *ip-precedence-value* **Or:** Ruijie(config-traffic-behavior)#**remark mpls-exp** *mpls-exp-value*	Sets the remark value of IPv4 packets.
Ruijie(config-traffic-behavior)#**remark ipv6 dscp** *dscp-value*	Sets the remark value of IPv6 packets.
Ruijie(config-traffic-behavior)#**remark mpls-exp** *mpls-exp-value*	Sets the remark value of MPLS packets.

Ruijie(config-traffic-behavior)#**remark cos** *cos-value*	Sets the 802.1Q remark value of Ethernet packet.
Ruijie(config-traffic-behavior)#**sub-traffic-policy** *traffic-policy-name*	Sets a sub-traffic policy.
Ruijie(config-traffic-behavior)#**rtp-shaping delay** *500* **clock-rate** *9000*	Set TRP shaping behavior.

Configuration example:

# Create a traffic behavior rule.

```
Ruijie(config)#traffic behavior tb1
Ruijie(config-traffic-behavior)#user-queue uq1 inbound
Ruijie(config-traffic-behavior)#service-class ef color green
Ruijie(config-traffic-behavior)#remark dscp 40
```

Configuring a Traffic Policy Rule

The traffic policy rule associates traffic classes and traffic behaviors, thus scheduling classified traffic according to user configurations. Configure this function as follows:

Command	Purpose
Ruijie(config)# **traffic policy** *traffic-policy-name*	Enters/Creates a traffic policy.
Ruijie(config-traffic-policy)# **classifier** *classifier-name* **behavior** *behavior-name* **precedence** *precedence-value*	Specifies a traffic behavior rule for a traffic class and sets the precedence. The smaller the value of the precedence, the higher the precedence is.

Configuration example:

# Create a traffic policy.

```
Ruijie(config)#traffic policy tp1
Ruijie(config-traffic-policy)#classifier tc1 behavior tb1 precedence 1
```

Applying a Traffic Policy to an Interface

Configure this function as follows:

Command	Purpose
Ruijie(config)# **interface** *interface-name*	Enters the interface configuration mode.
Ruijie(config-if)# **traffic-policy** *traffic-policy-name* **[inbound | outbound] [link-layer | all-layer]**	To apply a traffic policy, the layer parameter needs to be specified. Only Layer 3 policies and MPLS take effect by default. The link-layer parameter specified takes effect to only 802.1P Layer 2 packets. The all-layer parameter specified takes effect to both Layer 2 and Layer 3 packets. The link-layer and all-layer parameters can be specified for main interfaces only. The parameters take effect to the main interface and its associated sub-interfaces after specified. They cannot be specified for sub-interfaces.

Configuration example:

# Apply a traffic policy to an interface.

```
Ruijie(config)#int gigabitethernet 0/1
Ruijie(config-if-Gigabitethernet 0/1)#traffic-policy tp1 inbound
```

Configuring a CQ Template

Configuring an CQ template is to configure the scheduling mode (PQ, WFQ, and LPQ), traffic shaping value, and WRED parameters of eight kinds of PQs. PQ scheduling is performed on queues with the priority as ef, cs6, or cs7. WFQ scheduling is performed on queues with the priority as be, af1, af2, af3, or af4. Traffic shaping is not performed by default and the default discarding policy is tail discarding. You can configure multiple CQ templates and apply them to different interfaces. CQ scheduling takes effect on outbound traffic only.

Configure this function as follows:

Command	Purpose

Ruijie(config)# **port-queue** *port-queue-template-name*	Enters/Creates a CQ template.
Ruijie(config-port-queue)# **queue** *cos-value* {**pq** | **wfq weight** *weight-value* | **lpq**} [**shaping** *shaping-value*] [**wred** *wred-name*] [**depth** *depth-value*]	Sets CQ scheduling parameters.
Ruijie(config-port-queue-template)#**exit**	Exits the CQ template configuration.

Configuration example:

# Create a CQ template and set CQ scheduling parameters.

```
Ruijie(config)#port-queue pqt1
Ruijie(config-port-queue)# queue be lpq outbound
Ruijie(config-port-queue)# queue af1 wfq weight 10 shaping 100000 wred pwt1
Ruijie(config-port-queue)# queue cs7 pq shaping wred pwt1
Ruijie(config-port-queue)#exit
```

Applying the CQ to an Interface

The CQ applied to an interface takes effect on outbound traffic only.

Configure this function as follows:

Command	Purpose
Ruijie(config)# **interface** *interface-name*	Enters the interface configuration mode.
Ruijie(config-if)# **port-queue** *port-queue-template-name* [**shaping** *shaping-value*]	Applies the CQ to an interface.

Configuration example:

# Apply the CQ to an interface.

```
Ruijie(config)#int gigabitethernet 0/1
Ruijie(config-if-Gigabitethernet 0/1)#port-queue pqt1
```

Applying Resource Reservation Queue to Interface

The resource reservation queue applied to the interface is scheduled for outbound traffic only.

Run the following commands to configure this function:

Command	Purpose
Ruijie(config)# **interface** *interface-name*	Enters the interface configuration mode.
Ruijie(config-if)# **port-res-queue** *shaping-value* [**depth** *depth-value*]	Applies the resource reservation queue to the interface.

Configuration example:

#Apply the class queue to the interface.

```
Ruijie(config)#int gigabitethernet 0/1
Ruijie(config-if-Gigabitethernet 0/1)# port-res-queue 5000 depth 300
```

Note

The resource reservation queue takes effect only after the **port-queue** command is run on the interface.
The shaping value for the resource reservation queue is allocated form the shaping value configured in the **port-queue** command on the interface. The shaping value is automatically allocated only when there is a flow passing the resource reservation queue. When the flow stops for 3 minutes, the shaping value allocated to the resource reservation queue is invalidated.

Configuring HQoS Policy for AP Scenario

The bandwidth committed by the operator is configured for an AP member. The HQoS distributes load according to the physical link bandwidth corresponding to an AP member when scheduling traffic between AP members for routing. The physical bandwidths of a gigabit interface and a 10-gigabit interface are respectively 1000 Mbps and 10,000 Mbps. However, though the bandwidth allocated by the operator may be less than the committed bandwidth, the configuration is still performed according to the bandwidth committed by the operator. In this way, the HQoS schedules traffic for routing based on the actual available bandwidth of the egress, so that traffic on the operator links is controllable.

Run the following commands to configure this function:

Command	Purpose
Ruijie(config)# **interface** *interface-name*	Enters the interface configuration mode.
Ruijie(config-if)# **port-bandwidth** *bandwidth-value1*	Configures the maximum available bandwidth committed by the operator for the current AP member of traffic management.

Configuration example:

#Configure 100 Mbps available bandwidth on the link corresponding to Interface Gi0/0.

```
Ruijie(config)#int GigabitEthernet 0/0
Ruijie(config-if-GigabitEthernet 0/0)#port-bandwidth 100000
```

Note This command is available only on AP members.

An AP is configured to schedule traffic for routing. The AP is a logical port, and does not forward traffic. All traffic is forwarded by the physical member port. When the HQoS policy is applied to the AP, the HQoS policy schedules traffic between the AP members by the source or destination IP address for routing.

Run the following commands to configure this function:

Command	Purpose
Ruijie(config)# **interface AggregatePort-L3** *interface-name*	Enters the layer 3 AP configuration mode.
Ruijie(config-if-AggregatePort-L3 1)# **traffic-balance** { dst-ip | src-ip}	Schedules traffic between AP members by the source or destination IP address.

Configuration example:

#Perform traffic managment on the layer 3 AP by the source IP address.

```
Ruijie(config)#int AggregatePort-L3 1
Ruijie(config-if-AggregatePort-L3 1)#traffic-balance src-ip
```

Note This command is available only on the AP.

Configuring a Static VoQ Credit Point

By default, the VoQ credit point of a device is calculated dynamically based on the processing capability of the device, FAP bandwidth, and flow control. This configuration is to specify a static credit point for the destination device, so that VoQ requests are not updated dynamically during VoQ scheduling.

Configure this function as follows:

Command	Purpose
Ruijie(config)# **credit** {*device-id* *credit-value*}	Specifies the VoQ credit point for the destination device.

Configuration example:

# Set the VoQ credit point of the destination device with the ID as 4 to 10,000.

```
Ruijie(config)#credit 4 10000
```

Configuring System VoQ Scheduling

By default, system VoQ scheduling is disabled.

Configure this function as follows:

Command	Purpose
Ruijie(config)# **voq enable**	Enables system VoQ scheduling.

Configuration example:

# Enable system VoQ scheduling.

```
Ruijie(config)#voq enable
```

Clearing Queue Statistics

Command	Function
clear port-queue statistics interface *interface-name*	Clears CQ statistics.
clear user-queue statistics ***user-queue-name*** **{inbound | outbound}** devid *devid*	Clears UQ statistics.
clear user-group-queue statistics *user-group-queue-name* **{inbound | outbound} devid** *devid*	Clears GQ statistics.
clear voq statistics *class-queue-priority* **devid** *devid*	Clears VoQ statistics.

Showing Configurations

Command	Function
show wred [*wred-name*]	Displays WRED configuration information on the system.
Show flow-queue [*flow-queue-name*]	Displays FQ configuration information
show user-queue statistics *user-group-queue-name* **{inbound | outbound} [devid** ***devid*****]**	Displays UQ statistics.
show user-group-queue statistics *user-group-queue-name* **{inbound | outbound} [devid** *devid*]	Displays GQ statistics on the system.
show port-queue [*port-queue-name*]	Displays port-queue configuration information on the system.
show port-queue statistics [interface *interface*]	Displays port-queue statistics on the system.
show credit status [devid *devid*]	Displays credit system information.

Configuration example:

# Display the port-queue information on a system interface.

```
Ruijie# show port-queue interface gigabitethernet 1/1/1
[be]
 Pass:     42900556 packets,   2745666258 bytes
 Drop:            0 packets,            0 bytes
 Que :            0 packets,            0 bytes,      2073046 balance,           0 token
[af1]
 Pass:     43401132 packets,   2608782540 bytes
 Drop:            0 packets,            0 bytes
 Que :            0 packets,            0 bytes,        8960 balance,           0 token
[af2]
 Pass:     45091586 packets,   2707371120 bytes
 Drop:            0 packets,            0 bytes
 Que :            0 packets,            0 bytes,      2069592 balance,           0 token
[af3]
 Pass:     43496828 packets,   2613966540 bytes
 Drop:            0 packets,            0 bytes
 Que :            0 packets,            0 bytes,      2092532 balance,           0 token
[af4]
 Pass:     45170464 packets,   2711553720 bytes
 Drop:            0 packets,            0 bytes
 Que :            0 packets,            0 bytes,      2092532 balance,           0 token
[ef]
 Pass:     45099831 packets,   2708775960 bytes
 Drop:            0 packets,            0 bytes
 Que :            0 packets,            0 bytes,           0 balance,           0 token
[cs6]
 Pass:     46002386 packets,   2761254360 bytes
 Drop:            0 packets,            0 bytes
 Que :            0 packets,            0 bytes,           0 balance,           0 token
[cs7]
```

```
 Pass:     41955096 packets,   2520579480 bytes
 Drop:            0 packets,            0 bytes
 Que :            0 packets,            0 bytes,           0 balance,           0 token
```

Typical HQoS Configuration Examples

Configuration Example 1

Networking Requirement

- Device requirement

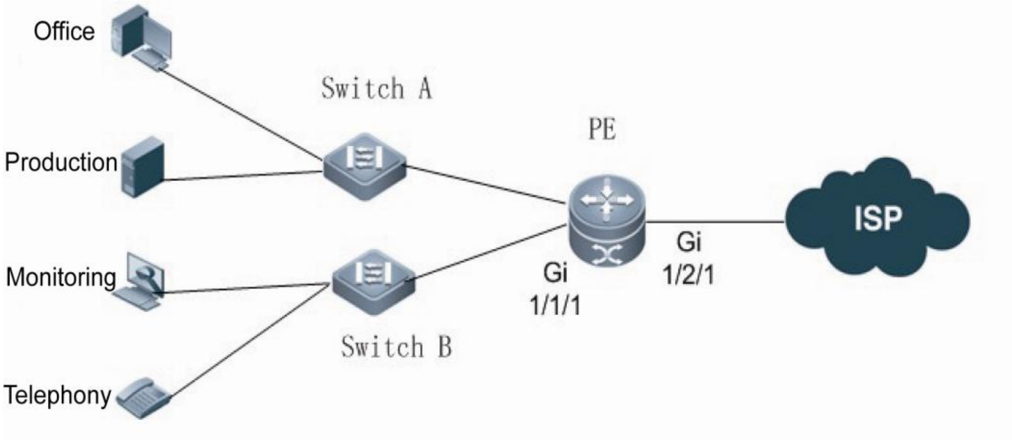
Two Ethernet switches and one router

- Configuration requirement

Each switch is connected to one user to provide two services. For user A, the CIR is 60 Mbit/s and peak bandwidth is 80 Mbit/s. For user B, the CIR is 30 Mbit/s and peak bandwidth is 40 Mbit/s. Both users are bound to a user group with the CIR as 90 Mbit/s and peak bandwidth as120 Mbit/s. Four services are identified by four IP addresses, that is, 10.1.10.1, 10.1.10.2, 10.1.20.1, and 10.1.20.2.

Networking Topology

Figure 3 HQoS Access Networking



Configuration Points

Configure traffic classification rules to distinguish between services.

Configure traffic behavior rules to perform traffic shaping on UQ and GQ.

Configuration Steps

Configure traffic classification rules.

```
# Configure traffic behavior rule tc1.
Ruijie(config)#access-list 100 permit ip 10.1.10.0 0.0.0.255  any
Ruijie(config)#traffic classifier tc1
Ruijie(config-traffic-classifier)#if-match acl 100
Ruijie(config-traffic-classifier)#exit
# Configure traffic behavior rule tc2.
Ruijie(config)#access-list 110 permit ip 10.1.20.0 0.0.0.255 any
Ruijie(config)#traffic classifier tc2
Ruijie(config-traffic-classifier)#if-match acl 110
Ruijie(config-traffic-classifier)#exit
```

Configure an FQ WRED template.

```
Ruijie(config)#wred-template wt1
Ruijie(config-wred-template)#color green low-limit 40 high-limit 60 discard-percent 10
Ruijie(config-wred-template)#color yellow low-limit 30 high-limit 50 discard-percent 10
Ruijie(config-wred-template)#color red low-limit 20 high-limit 40 discard-percent 10
Ruijie(config-wred-template)#exit
```

Configure FQ scheduling parameters.

```
Ruijie(config)#flow-queue-template fqt1
Ruijie(config-flow-queue-template)# queue be lpq
Ruijie(config-flow-queue-template)# queue af1 wfq weight 10 shaping 100 flow-wred wt1
Ruijie(config-flow-queue-template)# queue cs7 pq shaping shaping-percentage 20 flow-wred wt1
Ruijie(config-flow-queue-template)#exit
```

Configure an FQ mapping template.

```
Ruijie(config)#flow-mapping-template fmt1
Ruijie(config-flow-mapping-template)# map flow-queue af1 to port-queue ef
```

Configure a GQ.

```
Ruijie(config)#user-group-queue ugq1 inbound
Ruijie(config-user-group-queue)#shaping 120000
Ruijie(config-user-group-queue)#exit
```

Configure a UQ.

```
# Configure UQ uq1.
Ruijie(config)#user-queue uq1 inbound
Ruijie(confg-user-queue)#cir 60000 pir 80000
Ruijie(config-user-queue)#flow-queue-template fqt1
Ruijie(config-user-queue)#flow-mapping-template fmt1
Ruijie(config-user-queue)#user-group-queue ugq1
Ruijie(config-user-queue)#exit
# Configure UQ uq2.
Ruijie(config)#user-queue uq2 inbound
Ruijie(config-user-queue)#cir 30000 pir 40000
Ruijie(config-user-queue)#flow-queue-template fqt1
Ruijie(config-user-queue)#user-group-queue ugq1
Ruijie(config-user-queue)#exit
```

Configure traffic behavior rules.

```
# Configure traffic behavior rule tb1.
Ruijie(config)#traffic behavior tb1
Ruijie(config-traffic-behavior)#user-queue uq1 inbound
Ruijie(config-traffic-behavior)#exit
# Configure traffic behavior rule tb2.
Ruijie(config)#traffic behavior tb2
Ruijie(config-traffic-behavior)#user-queue uq2 inbound
Ruijie(config-traffic-behavior)#exit
```

Configure a traffic policy.

```
Ruijie(config)#traffic policy tp1
Ruijie(config-traffic-policy)#classifier tc1 behavior tb1
Ruijie(config-traffic-policy)#classifier tc2 behavior tb2
Ruijie(config-traffic-policy)#exit
```

Apply the policy to an interface.

```
Ruijie(config)#int gigabitethernet 1/1/1
Ruijie(config-if-Gigabitethernet1/1/1)#traffic-policy tp1 inbound
```

Configure a CQ WRED template.

```
Ruijie(config)# port-wred-template pwt1
Ruijie(config-port-wred-template)#color green low-limit 40 high-limit 60 discard-percent 10
Ruijie(config-port-wred-template)#color yellow low-limit 30 high-limit 50 discard-percent 10
Ruijie(config-port-wred-template)#color red low-limit 20 high-limit 40 discard-percent 10
Ruijie(config-port-wred-template)#exit
```

Configure a CQ.

```
Ruijie(config)#port-queue-template pqt1
Ruijie(config-port-queue-template)# queue be lpq outbound
Ruijie(config-port-queue-template)# queue af1 wfq weight 10 shaping 100000000 port-wred pwt1
Ruijie(config-port-queue-template)# queue cs7 pq shaping shaping-percentage 20 port-wred pwt1
Ruijie(config-port-queue-template)#exit
```

Apply the CQ to an interface.

```
Ruijie(config)#int gigabitethernet 1/2/1
Ruijie(config-if-Gigabitethernet 1/2/1)#port-queue pqt1 outbound
```

Verification

Display traffic policy statistics on an interface.

```
Ruijie#show user-queue statistics uq1 inbound devid 4
Ruijie#show user-queue statistics uq2 inbound devid 4
```

Display CQ statistics on an interface.

```
Ruijie# show port-queue interface gigabitethernet 1/2/1
```

Configuration Example 2

Networking Requirement

- Device requirement

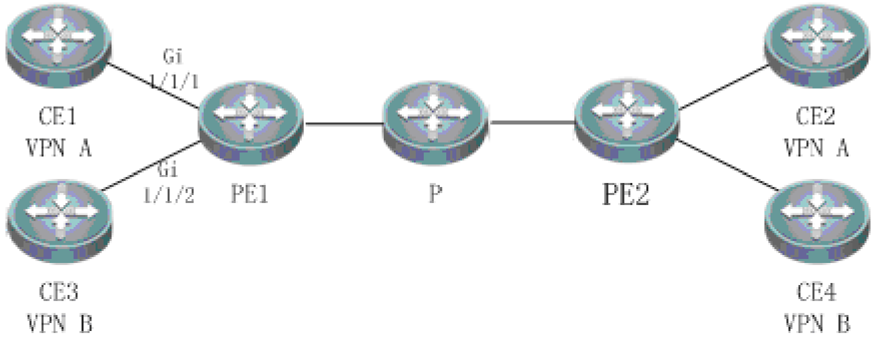
Seven routers

- Configuration requirement

Three routers constitute an MPLS backbone network. Two routers provide user access for VPN A, and two routers provide users access for VPN B. On each of the networks connected to CE1 and CE3, there is a multicast source with the multicast address as 232.0.0.1. For CE1, the CIR is 60 Mbit/s and peak bandwidth is 80 Mbit/s. For CE2, the CIR is 30 Mbit/s and peak bandwidth is 40 Mbit/s.

Networking Topology

Figure 4 HQoS Multicast Networking



Configuration Points

Configure traffic classification rules to distinguish between multicast addresses.

Configure traffic behavior rules to restrict the bandwidth of uplink multicast UQ.

Configuration Steps

Configure traffic classification rules.

```
# Configure traffic classification rule tc1.
Ruijie(config)#access-list 100 permit ip any host 232.0.0.1
Ruijie(config)#traffic classifier tc1
Ruijie(config-traffic-classifier)#if-match acl 100
Ruijie(config-traffic-classifier)#exit
# Configure traffic classification rule tc2.
Ruijie(config)#traffic classifier tc2
Ruijie(config-traffic-classifier)#if-match acl 100
Ruijie(config-traffic-classifier)#exit
```

Configure a UQ.

```
# Configure UQ uq1.
Ruijie(config)#user-queue uq1 inbound
Ruijie(confg-user-queue)#cir 60000 pir 80000
Ruijie(config-user-queue)#exit
# Configure UQ uq2.
```

```
Ruijie(config)#user-queue uq2 inbound
Ruijie(config-user-queue)#cir 30000 pir 40000
Ruijie(config-user-queue)#exit
```

Configure traffic behavior rules.

```
# Configure traffic behavior rule tb1.
Ruijie(config)#traffic behavior tb1
Ruijie(config-traffic-behavior)#user-queue uq1 inbound
Ruijie(config-traffic-behavior)#exit
# Configure traffic behavior rule tb2.
Ruijie(config)#traffic behavior tb2
Ruijie(config-traffic-behavior)#user-queue uq2 inbound
Ruijie(config-traffic-behavior)#exit
```

Configure a traffic policy.

```
Ruijie(config)#traffic policy tp1
Ruijie(config-traffic-policy)#classifier tc1 behavior tb1
Ruijie(config-traffic-policy)#exit

Ruijie(config)#traffic policy tp2
Ruijie(config-traffic-policy)#classifier tc2 behavior tb2
Ruijie(config-traffic-policy)#exit
```

Apply the policy to an interface.

```
Ruijie(config)#int gigabitethernet 1/1/1
Ruijie(config-if-Gigabitethernet1/1/1)#traffic-policy tp1 inbound

Ruijie(config)#int gigabitethernet 1/1/2
Ruijie(config-if-Gigabitethernet1/1/2)#traffic-policy tp2 inbound
```

Verification

Display traffic policy statistics on an interface.

```
Ruijie# show traffic policy tp1

Ruijie# show traffic policy t
```

VDA Configuration Example

Networking Requirements

- Device requirement

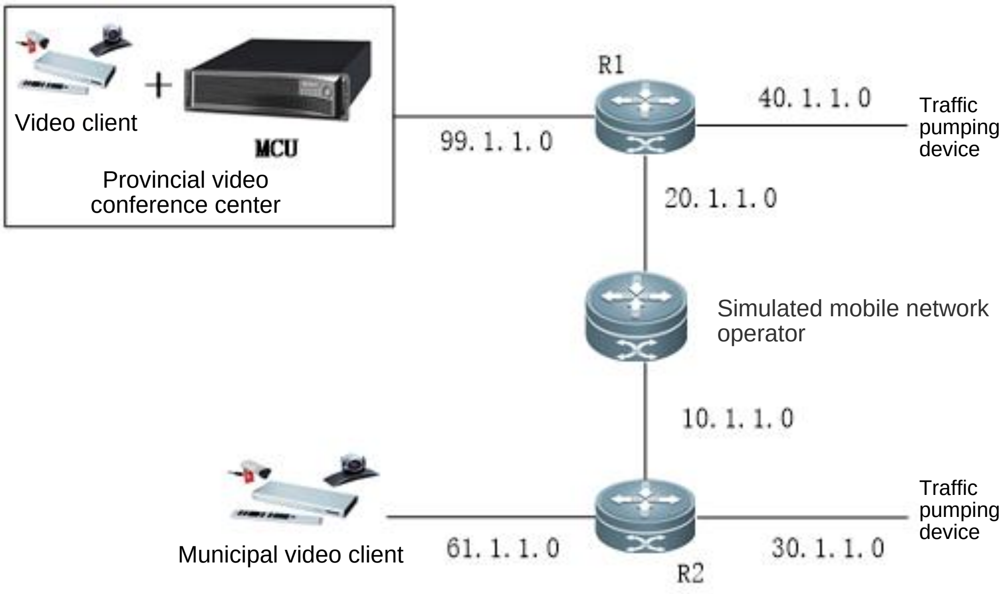
Three routers are required

- Configuration requirement

The provincial video conference center and the municipal video conference center are simulated. These two centers are connected via a 20 mbps dedicated line, to provide the video conference service and other production and office services. It is required to reserve bandwidth for a video conference and isolate the video conference service from other services. After the video conference ends, the reserved bandwidth is automatically released. Configurations in the two centers are the same, only the municipal egress device is described.

Network Topology

Figure 5 VDA Application



Configuration Key Points

1. Configure flow identification for the resource reservation queue.
2. Configure the resource reservation queue on an interface.

Configuration Steps

1. Configure the traffic classifier.

#Configure the traffic classifier for matching based on the interface and any ACL.

```
ip access-list extended 101
 10 permit ip any any
traffic classifier tc or
 if-match acl 101
```

2. Configure the traffic behavior.

#Configure the traffic behavior.

```
traffic behavior tb
 port-res-queue
```

3. Configure the traffic policy.

#Configure the traffic policy and the traffic behavior.

```
traffic policy tp
 classifier tc behavior tb precedence 1
```

4. Apply the traffic policy to an interface.

#Interface 0/0/2 is directly connected to the municipal conference center. Apply the traffic policy to this interface.

```
interface GigabitEthernet 0/0/2
 ip address 61.1.1.1 255.255.255.0
 traffic-policy tp inbound
```

5. Configure the resource reservation queue on the outbound interface.

#Connect Interface 0/0/1 (egress of the municipal network) to the operator device. Configure the resource reservation queue on this interface.

```
interface GigabitEthernet 0/0/1
 ip address 10.1.1.2 255.255.255.0
 port-queue pq shaping 20000
 port-res-queue 12000
```

Displaying

Display the user queue statistics on the device.

```
Ruijie# show port-queue statistics interface gi 0/0/1
```

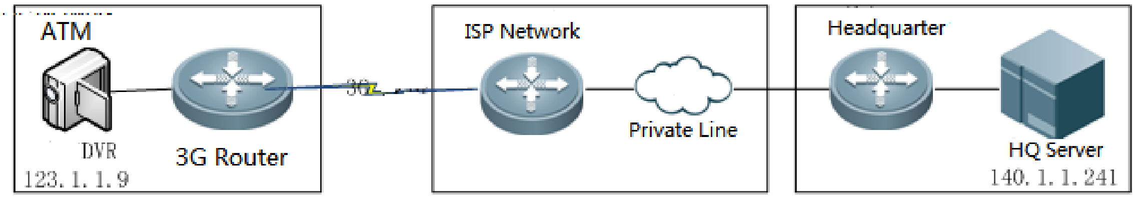
Examples for Configuring RTP Traffic Shaping

Networking Requirements

1. Off-bank ATMs adopt Ruijie routers as access routers that are connected to servers in headquarters via 3G circuit.
2. Monitoring videos are transmitted between routers of branches and the servers of the headquarters by application of TCP and converged by RSR77 in the headquarters.
3. Data flows are not encrypted or encapsulated by other tunnels before reaching the routers adopting optimizing strategies.

Network Topology

Figure 6 RTP Shaping Application Network Topology



Configuration Tips

1. Turn on WAN-TA function.
2. Identify characteristics of data flows to realize RTP business.
3. Configure RTP shaping behavior.

Configuration Steps

1. Transmitted optimization should be turned on when RSR77 of the headquarters are configured.

Configure video transmitted optimization to transit monitoring videos between routers and servers. ACL101configues RTP data flows from the branches to the headquarters. With video data flows of off-bank ATMs reaching the servers in the headquarters, the transmitted optimization of convergence routers is enabled. And wan-ta policy is applied on the in-coming interface gigaethernet2/1/1.

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)#ip access-list extended 101
Ruijie(config-ext-nacl)# 10 permit tcp any eq 554 any
Ruijie(config)#wan-ta enable
Ruijie(config)# wan-ta policy video
Ruijie(config-if-GigabitEthernet 2/1/1)#wan-ta-policy video list 101
```

2. RTP Traffic Shaping to RSR77 of the Headquarters

Configuring RTP Traffic Shaping to RSR77 of the Headquarters

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)#
Ruijie(config)#traffic classifier tc1 or
Ruijie(config-traffic-classifier)# if-match acl 101
Ruijie(config-traffic-classifier)#
Ruijie(config-traffic-classifier)#traffic behavior tb1
Ruijie(config-traffic-behavior)# rtp-shaping delay 500 clock-rate 89950
Ruijie(config)#traffic policy tp1
Ruijie(config-traffic-policy)# classifier tc1 behavior tb1 precedence 1
Ruijie(config-if-GigabitEthernet 2/1/1)#traffic-policy tp1 inbound
```

Verification

1. Displaying WAN-TA Policy Configuration

```
Ruijie#show wan-ta policy video
wan-ta policy: video
    Congestion Control : low-bandwidth-delay
    SACK Support: TRUE
    Initial Congest Window: 10 MSS
    Maxitum Segment Size: 1460
```

```
   Keepalvie Interval(retry): 120(9)

apply on interfaces:
interface name                      list
GigabitEthernet 2/1/1                 101
```

2. Displaying RTP-shaping Statistics

```
Ruijie#show rtp-shaping statistics
hqos_rtp_dump[185566fc][1]
rtp-que[184bdec0]:
       delay[1000],clock_rate[89950]
       flow[10.255.255.91,10.255.255.213,6,554,49419]
       que-pkts[0],pass_pkts[6050],drop_pkts[0],disorder_pkts[0],none_video_pkts[5692]
       last_rtp_time[3933490296],last_sys_time[2267149741]
```

Configuring IP Multicast

Configuring IP Multicast Routing

Overview

This chapter describes how to configure IPv4 multicast routing protocols. To obtain complete descriptions of multicast routing commands, see the "Multicast Routing Commands" section.

The traditional IP transmission only allows one host to transmit packets to a single host (unicast communication) or all hosts (broadcast communication). Multicast, however, allows one host to send the packets to some hosts (also known as group members).

The destination addresses of packets sent to the group member are Class-D network addresses (224.0.0.0–239.255.255.255). Multicast packets are UDP packets with best effort service. It does not provide reliable transmission and error control as TCP.

The multicast application consists of the sender and receiver. The sender can send multicast packets without needing to join a group. In contrast, the receiver can receive the multicast packets from the group only after joining the group.

Group members are dynamic. A host can join or leave a group at any time. Furthermore, there is no limit on the position or number of group members. A host can join more than one group simultaneously if necessary. Consequently, the active status and the number of members of a group vary from time to time.

Devices run a multicast routing protocol such as the Protocol-Independent Multicasting-Dense Mode (PIM-DM) and the Protocol-Independent Multicasting-Sparse Mode (PIM-SM) to maintain their routing tables to forward multicast messages, and use the Internet Group Management Protocol (IGMP) to learn the status of the members within a group on their directly attached subnets. A host can join or leave an IGMP group by sending IGMP Report messages.

IP multicast applies to one-to-many multimedia applications.

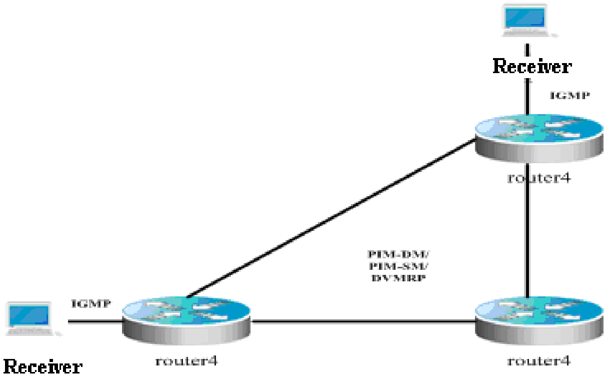
Implementation of IP Multicast Routing

Multicast routing protocols include:

- IGMP: Runs between the routers and the hosts to trace the relationship of group members.
- PIM-DM: A multicast routing protocol in dense mode, which runs between multicast devices to establish the multicast routing table for forwarding.
- PIM-SM: A multicast routing protocol in sparse mode, which runs between multicast devices to establish the multicast routing table for forwarding.
- DVMRP: Distance Vector Multicast Routing Protocol, which runs between multicast devices to establish the multicast routing table for forwarding.

Figure 1 shows IP multicast routing protocols used within the IP multicast environment.

Figure 1 IP Multicast Routing Protocols within the IP Multicast Environment



IGMP

To enable IP multicast, hosts and routers must support the IGMP protocol. This protocol is used by hosts to report their group memberships to multicast routers on the directly-connected network, allowing the multicast routers to determine how to forward multicast traffic.

By using the information obtained from IGMP, multicast routers create an interface-based multicast group member list. The list is activated only when at least one host on an interface is a member of the group. IGMPv1, IGMPv2 and IGMPv3 are currently supported.

IGMPv1

There are only two types of IGMP messages defined in IGMPv1:

- Membership query
- Membership report

A host sends a membership report to indicate that it is interest in joining a group, and the router sends membership queries periodically to ensure that the group has at least one host. When there is no hosts in that group, the device will delete it.

IGMPv2

In IGMPv2, there are only four types of IGMP messages:

- Membership query
- Version 1 membership report
- Version 2 membership report
- Leave group

IGMPv2 is basically the same as IGMPv1, except that IGMPv2 creates a Leave group message for hosts. For IGMPv2, hosts report leave messages to routers which then send queries to check whether there is a host in the multicast group. This makes joining and leaving a group more efficient.

In the multicast network running IGMP, a multicast router is dedicated for sending IGMP query messages. This router is called a querier which is selected through an election mechanism. At first, all routers are queriers. If a router receives a query message from another router with a lower IP address, it becomes a non-querier. Consequently, there is only one querier which has the lowest IP address among all multicast routers on the network.

If a querier is invalid, new querier will be elected. Non-queriers keep a timer for Other Querier Present Interval. Every time when a router receives a membership query packet, it resets the timer. If the timer expires, the router starts to send query messages and selects new querier again.

Queriers must periodically send membership queries to ensure that other routers on the network know that the querier still works. For this purpose, the querier maintains one query interval timer. When it sends membership query messages, this timer will be reset. When the interval timer times out, the querier sends another membership query.

When a new router appers, it sends a series of general query messages to solicit membership information. The number of general query packets depends on the Startup Query Count configured on the router. The initial general query interval is defined by the Startup Query Interval.

When a querier receives a leave group message from a host, it must send a group-specific membership query to see whether the host is the last one to leave the group. Before the querier stops forwarding packets to the group, it sends a series of such packets, the number of which is equal to the Last Member Query Count. The querier sends multiple group-specific membership queries to ensure that there is no member in the group. Such a query is sent every other the Last Member Query Interval seconds. When no response is received, the querier stops forwarding multicast packets to the group on the specified interface.

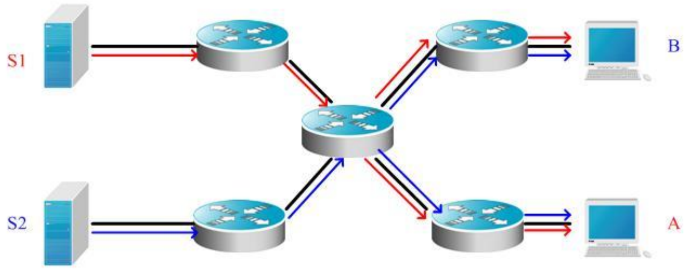
IGMPv3

Both IGMPv1 and IGMPv2 have the following defects:

- Lack of efficient measures to control multicast sources
- Difficult to establish multicast paths due to ignorance of multicast source locations.
- Difficult to find a unique multicast address. It is possible that multicast groups are using the same multicast address.

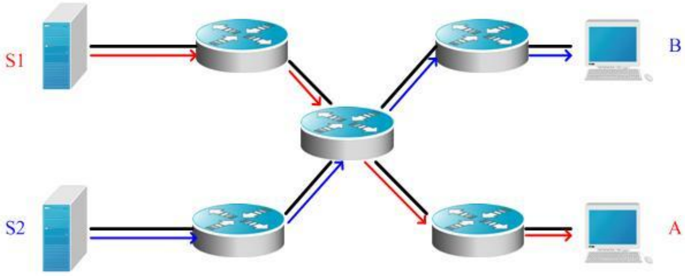
On the basis of IGMPv1 and IGMPv2, IGMPv3 provides an additional source filtering multicast function. In IGMPv1 or IGMPv2, hosts determines whether to join a group by group address and, once it joins the group, it receives multicast traffic forwarded from any source to that group address. In IGMPv3, hosts are enabled to report the multicast group they desire to join in and the multicast source from which they expect to receive traffic. A host specifies sources from which they want to receive multicast traffic through an INCLUDE list or an EXCLUDE list. Besides, IGMPv3 saves bandwidth by preventing unnecessary, illegal multicast data flows from occupying network bandwidth. It is particularly useful in the case where multiple multicast sources share one multicast address. IGMPv1 and IGMPv2 can also implement "source address filtering" in some sense, which, however, is performed on hosts receiving multicast traffic. As shown in the following diagram, two multicast sources (S1 and S2) send out traffic directed to the same multicast group address (G). This multicast traffic from S1 and S2 will be sent to all hosts receiving traffic from G. If host A only wants to receive multicast traffic from S1, it has to filter out traffic from S2 by running approriate client software.

Figure 2 Multicast traffic forwarded without source filtering



If multicast routers on the network support IGMPv3, host A wants to receive traffic from S1 only, it sends out an IGMPv3 packet in the form of “join G include S1”. host B wants to receive traffic from S2 only, it sends out an IGMPv3 packet in the form of “ join G include S2”. In this way, the traffic is forwarded as shown in Figure 3. This saves bandwidth.

Figure 3 Multicast traffic forwarded with source filtering



Based on IGMPv2, IGMPv3 adds the following two kinds of messages:

- Membership query
- Version 3 membership report

There are three types of membership query:

- General Query: used to learn information of all multicast members on an interface.
- Group-Specific Query: used to learn information of members of a specific group on an interface.
- Group-and-Source-Specific Query: a new type specified in IGMPv3 used to learn whether there is a member on an interface wants to receive group-specific multicast traffic from sources in the specified source list.

Membership report in IGMPv3 is different from that defined in IGMPv2. The IGMPv3 membership reports are always sent with an destination address of 224.0.0.22. Besides, an IGMPv3 membership report can contain one or more group records, containing a group address and an list of source addresses. Group records have the following types.

- IS_IN: indicates that the filter mode between a multicast group and the multicast source list is INCLUDE, that is, only multicast data sent from the specified multicast source list to the multicast group is received. If the specified multicast source list is empty, it indicates leaving the multicast group, which is equivalent to the leave packet in IGMPv2.
- IS_EX: indicates that the filter mode between a multicast group and the multicast source list is EXCLUDE, that is, only multicast data sent from the multicast sources not included in the specified multicast source list to the multicast group is received.
- TO_IN: indicates that the filter mode between a multicast group and the multicast source list is changed from EXCLUDE to INCLUDE.
- TO_EX: indicates that the filter mode between a multicast group and the multicast source list is changed from INCLUDE to EXCLUDE.

- ALLOW: indicates that multicast data from certain multicast sources are allowed. If the current relationship is INCLUDE, these multicast sources are added to the existing multicast source list. If the current relationship is EXCLUDE, these multicast sources are deleted from the existing multicast source list.
- BLOCK: indicates that multicast data from certain multicast sources are prohibited. If the current relationship is INCLUDE, these multicast sources are deleted from the existing multicast source list. If the current relationship is EXCLUDE, these multicast sources are added to the existing multicast source list.

For compatibility consideration, IGMPv3 can identify packets of IGMPv1 and IGMPv2.

PIM-DM

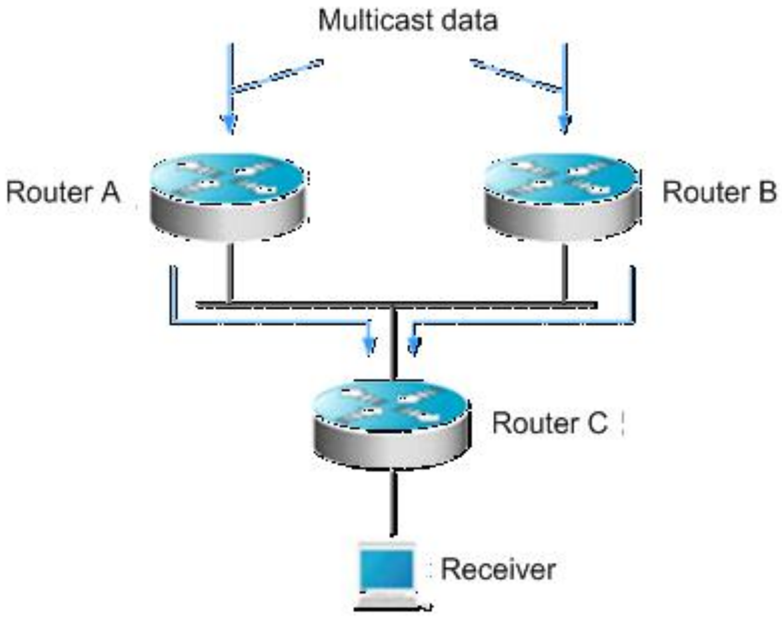
Protocol Independent Multicast-Dense Mode (PIM-DM) is a dense-mode multicast routing protocol, which is suitable for small-sized networks with densely distributed multicast members. As PIM-DM does not rely on any specific unicast routing protocol, it is called protocol independent multicast routing protocol. PIM-DM is defined in RFC 3973.

PIM-DM devices discover neighbors through Hello messages. After startup, a PIM-DM device sends a Hello message to each PIM-DM enabled interface periodically. The Hello message has a field of **Hello Hold Time**, which defines the maximum duration that a neighbor waits for the next message. If the neighbor does not receive another Hello message from the sender within this duration, this device will be removed from the adjacency list.

PIM-DM builds a shortest path tree (SPT) through flood and prune. PIM-DM assumes that when a multicast source begins to send a multicast packet, all the systems in the network need to receive this packet. As a result, this packet is forwarded to every system. The reverse path forwarding (RPF) check is performed for the packets received from the upstream interface. Those packets that fail to pass the check will be discarded. For the packets passing the check, the outgoing interface is computed based on the (S, G) pair of the packets, that is, source address and group address. If the outgoing interface is not null, an outgoing interface entry is created from the (S, G) pair and the multicast packet is forwarded through this outgoing interface. If the computed outgoing interface is null, a prune message is sent to the RPF neighbor, informing the upstream neighbor not to forward the multicast packets from the (S, G) pair to this interface. After the prune message is received on the upstream interface, the device marks the sending interface as pruned state and sets a pruned state timer. In this way, an SPT is created with the multicast source as its root.

PIM-DM uses the Assert mechanism to eliminate redundant routes.

Figure 4 Assert mechanism of PIM-DM



As shown in Figure-4, the multicast data arrives at Routers A and B at the same time, which forward the data to Router C. In this case, Router C receives duplicated data, which is not allowed. So there must be a mechanism to select Router A or B to forward the multicast data to Router C. This is the Assert mechanism of PIM-DM.

PIM-DM uses the state refresh message to update network state. The device directly connected to the multicast source sends the state refresh message to the downstream devices periodically to advertise the network topology changes. The devices receiving the message add their topology state to the state refresh message by modifying some fields, and then send it to the downstream devices. When the refresh message arrives at the leaf devices, the entire network state is updated.

PIM-DM uses the Graft mechanism to reestablish the connection with upstream devices. If the network topology of a downstream device in pruned state changes and the device needs to receive multicast data from a (S, G) pair, it sends the graft message to the upstream device. Upon receiving the graft message, the upstream device responds with a Graft-Ack message and forwards the multicast data to the downstream device again.

DVMRP

The Distance Vector Multicast Routing Protocol (DVMRP) is the first widely used multicast routing protocol in the Internet. It is also in dense mode. Like PIM-DM, DVMRP also uses the reverse path multicast mechanism to establish a distribution tree to forward multicast packets. The difference between the two protocols is that PIM-DM does not rely on specific unicast routing protocols and DVMRP replies on RIP.

A DVMRP device advertises itself, learns neighbor addresses and establishes adjacency through Probe packets. The DVMRP device establishes the adjacency relationship when the Probe packet received from a neighbor contains the IP address of the DVMRP device.

DVMRP neighbors exchange source route information by periodically sending Report packets. The information includes the source network mask and hop count. Such information is stored in the DVMRP routing table that is independent of the unicast routing table and used for RPF check during source tree creation.

DVMRP is also a multicast routing protocol in dense mode and creates SPTs for each multicast source. Initial multicast traffic is forwarded along the entire SPT, but DVMRP does not forward redundant paths. For the specified SPT, the device will send Prune packets to the upstream device after acknowledging that it does not need to receive multicast traffic. The device does not need to receive specified multicast traffic if no downstream neighbor exists and no multicast member information exists. As DVMRP is a multicast routing protocol in dense mode, multicast traffic is redistributed once pruning times out.

In addition, to enable the multicast receiver to join the SPT quickly, DVMRP supports the Graft and Graft-Ack mechanisms. The Graft mechanism adds the pruned paths to the SPT quickly, while the Graft-Ack mechanism avoids loss of Graft messages due to busy networks.

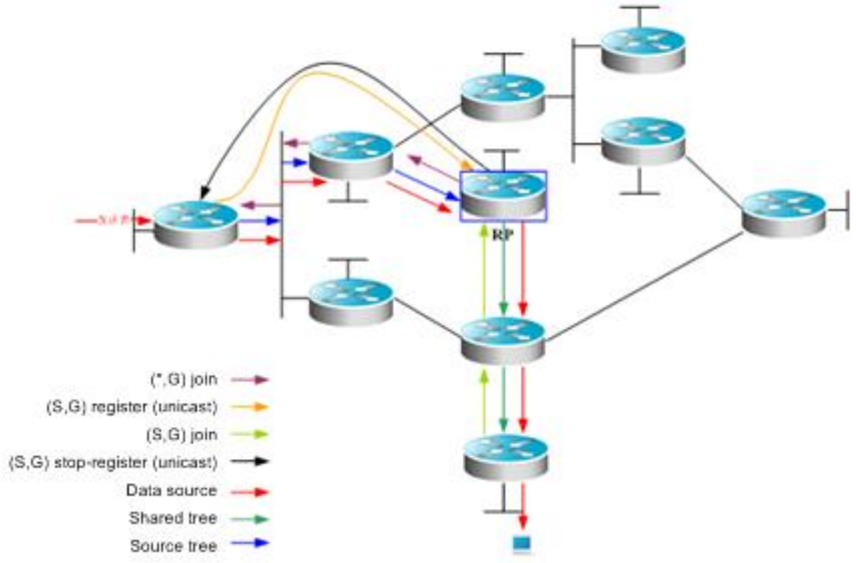
This product supports the complete DVMRP protocol.

PIM-SM

The Protocol Independent Multicast (PIM) is designed by the Inter-Domain Multicast Routing (idmr) working group. As its name implied, PIM does not rely on any specific unicast routing protocol. It can use a unicast routing table established by any unicast routing protocol to perform the RPF check function, instead of maintaining separate multicast routing tables to implement multicast forwarding. As PIM is not required to receive or distribute route updates, compared to other multicast routing protocols, it costs much less. PIM is designed to support shortest path trees (SPTs) and rendezvous point trees (RPTs) simultanenously and enable flexible conversion between them, so that their advantages can be used to improve multicast efficiency. There are two PIM modes: dense mode and sparse mode.

The Protocol Independent Multicast – Sparse Mode (PIM-SM) is a multicast routing protocol of sparse mode. In a PIM-SM domain, the PIM-SM-enabled device periodically sends Hello messages to discover adjacent PIM-SM devices and selects a designated router (DR) in a multi-access network. The DR is responsible for sending Join/Prune messages towards the root of themulticast distribution tree from its directly connected group member, or itsdirectly connected multicast source.

Figure 5 Explicit join mechanism of PIM-SM



PIM-SM forwards multicast data packets by establishing a multicast distribution tree. The multicast distribution tree is divided into two types: Shared Tree that takes the RP of the group G as the root and Shortest Path Tree that takes the multicast source as the root. PIM-SM establishes and maintains the multicast distribution tree by use of the explicit join/prune mechanism. As shown in Figure-5,

When the DR at the receiving end receives a report packet from the receiving end, it sends a (*.G)join packet towards the RP of group G to join the shared tree.

When the DR at the data source receives multicast data from the source host, it encapsulates the multicast data into a register message and unicasts it to the RP. Then the RP will forward the decapsulated data packets to group members along the shared tree.

The RP sends an (S, G)join packet to the first-hop device in the source direction to join the shortest path tree of this source. In this way, the source’s packets are sent to the RP without encapsulation along its shortest path tree.

When the first multicast data reaches along the SPT, the RP sends the stop-register message to the DR at the source, notifying the DR of stopping register encapsulation. Afterwards, the DR at the source does not encapsulate register packets but sends them to the RP along its shortest path tree, which then forwards the packets to group members along the shared tree. When no multicast data is required, the DR at the receiving end multicasts the prune message to group G’s RP hop by hop to prune the shared tree.

PIM-SM also offers a mechanism of selecting the root point (RP). One or more Candidate-BSRs are configured in a PIM-SM domain. PIM-SM selects a BSR by following a certain rule. There are also Candidate-RPs in a PIM-SM domain that unicast the packets including their IP addresses and available multicast groups to the BSR. The BSR will periodically generate a BSR message which includes a series ofcandidate RPs and corresponding multicast group addresses. The BSR messages are sent hop-by-hop within the entire domain. The device receives and saves these BSR messages. If the DR receives a report on the member relationship of a multicast group from its directly connected host but has no route entries of the multicast group, the DR will use a Hash algorithm to map the multicast group address to a candidate RP that can serve this group. Then, the DR multicasts the Join/Prune message to the RP hop-by-hop. If the DR receives multicast data packets from its directly connected host but has no route entries of the multicast group, the DR will usea Hash algorithm to map the multicast group address to a candidate RP that can serve this group. Then the DR encapsulates multicast data packets into a register message and unicasts it to the RP.

The main difference between PIM-SM and the flood/prune model-based PIM-DM is that PIM-SM is based on the explicit join model. In other words, the receiver sends the join message to the RP, while the router only forwards the packets of that multicast group on the outgoing interface that has joined a multicast group. PIM-SM uses the shared tree to forward multicast packets. Each group has a Rendezvous Point (RP). The multicast source sends data to the RP along the shortest path, and then the RP sends the data to the receivers along the shortest path. This is similar to CBT, but PIM-SM does not use the concept of core. One of the major advantages of PIM-SM is that it not only receives multicast messages through the shared tree but also provides a shared tree-to-SPT conversion mechanism. Such conversion reduces network delay and possible congestion on the RP, but it consumes enormous router resources. So it is suitable for the case where there are only a few multicast data sources and network groups.

PIM-SM uses the shared tree and SPT to distribute multicast frames. At this time, it is assumed that other devices don't want to receive these multicasts unless otherwise stated definitely. When a host joins a group, the equipment connected to the host must notify the root (or the RP) by using the PIM join message. This join message is transferred one after another through the routers to create a shared tree structure. Therefore, the RP records the transfer path and also the register message from the first hop router (DR) of the multicast source, and improves the shared tree upon these two messages. The branch/leaf messages are updated by periodically querying messages. With the shared tree, the multicast source first sends multicast packets to the RP, guaranteeing that all the receivers can receive them. The notation (*.G) represents a tree. The asterisk (*) represents all sources and G represents a specific multicast address. The prune message is also used in the shared tree. That is, the branch/leaf will send prune messages once it is not expecting to receive multicast frames.

PIMv2 BSR is a method of distributing group-to-RP messages to all devices without the need of setting an RP for them. BSR distributes mapping information by propagating BSR messages hop by hop. At first, BSR is selected among routers in the same process as selecting a root bridge based on priority level among layer 2 bridges. Each BSR checks the BSR messages and only forwards those having a priority higher than or equal to its own (higher IP address). The selected BSR sends its BSR message to the all-PIM-routers multicast group (224.0.0.13), where TTL is 1. After the adjacent PIMv2 router receives the message, it multicasts it while setting the TTL to 1. In this way, the BSR message is received by all devices hop by hop. Since the message contains the IP address of the BSR, the candidate BSR can know which router is the current BSR based on this message. The candidate RPs send candidate RP advertisements to announce in which address ranges they can become an RP. The BSR stores them in its local candidate RP cache. The BSP notifies all PIM routers of its local candidate RPs periodically. These messages reach various devices hop by hop in the same way.RPF Check

Multicast routing protocols depend on existing unicast route messages, MBGP routes or static multicast routes to create multicast routing entries. When creating multicast routing entries, multicast routing protocols run the Reverse Path Forwarding (RPF) check mechanism to ensure that multicast packets are transmitted along proper paths while avoiding loops.

RPF check is on the basis of unicast routes, MBGP routes or static multicast routes.

- The unicast routing table summarizes the shortest paths to each destination network segment.
- The MBGP routing table directly offers multicast routes.
- The multicast static routing table lists the RPF route messages that the user configures statically and manually.

RPF Check Process

The multicast routing protocol searches the unicast routing table, the MBGP routing table and the static multicast routing while performing an RPF check. The process is as follows:

1) First of all, select an optimal route from the unicast routing table, the MBGP routing table and the static multicast routing table, respectively.

- Select an optimal route from the unicast routing table for RPF check:
  - Use the IP address of the packet source as the destination address to search the unicast routing table and select an optimal unicast route.
  - If the unicast route has only one next hop, check whether multicast is enabled on the egress of the next hop.
    - If no, the unicast route is not suitable for RPF check.
    - If yes, the unicast route is suitable for RPF check and the egress serves as the RPF interface.
  - If the unicast route has more than one next hop, traverse all next hops and check whether multicast is enabled on the egress of one next hop.
    - If no, traverse the next hop.
    - If yes, the unicast route is suitable for RPF check and the egress serves as the RPF interface.
    - If no multicast is enabled on the egress after all next hops are traversed, there is no unicast route suitable for RPF check.
  - If no optimal route exists, there is no unicast route suitable for RPF check.
- Select an optimal route from the MBGP routing table for RPF check:
  - Use the IP address of the packet source as the destination address to search the MBGP routing table and select an optimal MBGP route.
  - The MBGP route has only one next hop. Check whether multicast is enabled on the egress of the next hop.
    - If no, the MBGP route is not suitable for RPF check.
    - If yes, the MBGP route is suitable for RPF check.
  - If no optimal route exists, there is no MBGP route suitable for RPF check.
- Select an optimal route from the static multicast routing table for RPF check:
  - Use the IP address of the packet source as the destination address to search the static multicast routing table and select an optimal static multicast route.
  - If the static multicast route has only one next hop, check whether multicast is enabled on the egress of the next hop.
    - If no, the static multicast route is not suitable for RPF check.
    - If yes, further check whether there is a unicast route available for RPF check.
      - If the next hop does not associate with the unicast protocol number, the static multicast route is suitable for RPF check.
      - If there is no unicast route available for RPF check, the static multicast route is suitable for RPF check.
      - If there is a unicast route available for RPF check, but the unicast protocol number is inconsistent with the one associated with the next hop of the static multicast route, the static multicast route is not suitable for RPF check.
      - If there is a unicast route available for RPF check, and the unicast protocol number is consistent with the one associated with the next hop of the static multicast route, the static multicast route is suitable for RPF check.
  - If the static multicast route has more than one next hop, traverse all next hops and check whether multicast is enabled on the egress of one next hop.
    - If no, traverse the next hop.
    - If yes, further check whether there is a unicast route available for RPF check.
      - If the next hop does not associate with the unicast protocol number, the static multicast route is suitable for RPF check. The outbound interface is the RPF interface.

- If there is no unicast route available for RPF check, the static multicast route is suitable for RPF check. The outbound interface is the RPF interface.
- If there is a unicast route available for RPF check, but the unicast protocol number is inconsistent with the one associated with the next hop of the static multicast route, traverse the next hop.
- If there is a unicast route available for RPF check, and the unicast protocol number is consistent with the one associated with the next hop of the static multicast route, the static multicast route is suitable for RPF check. The outbound interface is the RPF interface.
- If no multicast is enabled on the egress after all next hops are traversed, there is no static multicast route suitable for RPF check.

■ If no optimal route exists, there is no static multicast route suitable for RPF check.

Select one from these three optimal routes for RPF check.

■ If the longest match routing rule is configured, select the longest match route; if these three routes are of the same mask, select the one of the highest priority; if they are of the same priority, select the one in the order of static multicast route, MBGP route and unicast route.

■ If the longest match routing rule is not configured, select the one of the highest priority; if they are of the same priority, select the one in the order of static multicast route, MBGP route and unicast route.

⚠ Caution

The effectiveness of MBGP routes recurs on unicast routes rather than distance. The implementation of current MBGP protocols does not support equal-cost routes.

The effectiveness of static multicast routes recurs on unicast routes rather than distance.

If the static unicast route is selected as RPF route, the route must be configured with the next hop IP address. The PIM protocol will select RPF neighbors based on this next hop IP address. If the static unicast route is not configured with the next hop IP address, the PIM protocol cannot obtain RPF neighbors.

☑ For the commands including the VRF parameters, only the RSR20, RSR30, RSR50 and RSR50E devices support the VRF parameters.

Configuring IP Multicast Routing

Enabling IP Multicast Forwarding

Multicast data packets and protocol packets can be received and processed by related multicast protocols only after the multicast routing forwarding function is enabled.

Command	Function
Ruijie(config)# **ip multicast-routing** [**vrf** *vrf-name*]	Enables multicast routing forwarding. Enable multicast routing based on VRF if it is carried; Enable the default multicast routing globally if VRF is not carried.

Enabling IP Multicast Routing Protocols

Use the following commands to enable the IP multicast function on an interface.

Command	Function

Command	Function
Ruijie(config-if)# **ip pim dense-mode**	Enters the interface on which PIM-DM is to be enabled and enables PIM-DM in interface configuration mode. This command must be configured on a layer 3 interface.
Ruijie(config-if)# **ip pim sparse-mode**	Enters the interface on which PIM-SM is to be enabled and enables PIM-SM in interface configuration mode. This command must be configured on a layer 3 interface.
Ruijie(config-if)# **ip dvmrp enable**	Enters the interface on which DVMRP is to be enabled and enables DVMRP in interface configuration mode. This command must be configured on a layer 3 interface.

The following example shows how to configure PIM-DM on interface GE 0/3:

```
Ruijie(config)# ip multicast-routing
Ruijie(config)# interface  gigabitEthernet 0/3
Ruijie(config-if)# ip address 192.168.194.2 255.255.255.0
Ruijie(config-if)# ip pim dense-mode
```

Note

If IPv4 multicast routing is enabled globally, enabling IP multicast routing protocols on an interface will also enable the IGMP function. Only the IP multicast routing protocols of one mode can be enabled on one interface.

Caution

After the layer 3 multicasting is enabled on the Private VLAN and Super VLAN, if the multicast source exists in the sub-VLAN, one more route entry must be duplicated and the ingress is the sub-VLAN in which the multicast streams enter as the ingress validity check is required for multicast forwarding. As a result, one more multicast hardware entry is occupied, that is, multicast capacity is reduced by 1.

Enabling IGMP

The IGMP protocol is enabled with the enabling of IP multicast route forwarding and IP multicast routing protocols.

Configuring IP Multicast Routing

Configuring TTL Threshold

To limit the TTL of the data packets allowed to pass an interface, configure the TTL threshold.

Use the following command to configure the TTL threshold of multicast packets allowed to pass an interface. Use the **no** form of this command to restore the default value. The default value is **0**.

Command	Function

Command	Function
Ruijie (config-if) # **ip multicast ttl-threshold** *ttl-value*	Configures the TTL threshold of an interface. *ttl-value* ranges from 0 to 255.

Limiting the Number of Entries to Be Added to the IP Multicast Routing Table

Use the following command to limit the number of entries to be added to the IP multicast routing table in global configuration mode. Use the **no** form of this command to restore the default value. The default value is **1024**.

Command	Function
Ruijie (config) # **ip multicast route-limit** *limit* [*threshold*]	Limits the number of entries to be added to the IP multicast routing table. *limit* specifies the number of entries to be added to the multicast routing table. The range is from 1 to 2147483647. The default is 1024. *threshold* (optional) specifies the number of routes triggering alarm. The default is 2147483647.

Caution

As the hardware is limited for different models, the routes exceeding the hardware entry threshold need to be forwarded through software, resulting in performance degrade.

Configuring IP Multicast Boundary for a Specific IP Group

Use the following command to set IP multicast boundary for a specific IP group in interface configuration mode. Use the **no** form of this command to restore the default value.

Command	Purpose
Ruijie (config-if) # **ip multicast boundary** *access-list* [**in** | **out**]	Sets IP multicast boundary for a specific IP group. Numerical standard ACL or name can be used to specify an IP group.

This command filters the IGMP, PIM-SM and PIM-DM packets associated with the IP group. Multicast packets will not flow in or out through the multicast boundary interface.

Note

The ACL in this command supports matching of destination IP addresses, not group IP addresses or source IP addresses.

Configuring IP Multicast Static Route

IP static multicast route enables multicast packet forwarding through a path different from IP unicast path. RPF check is always performed for IP multicast packet forwarding. The real interface receiving packets is the expected one, namely the next hop interface of IP unicast route used to transmit to the sender. The check is reasonable when IP unicast topology is consistent with IP multicast topology. In some cases, however, it is better to make difference between IP unicast path and IP multicast path.

Static multicast route enables devices to execute RPF check according to configurations rather than the IP unicast routing table. Consequently, tunnel technology is used for IP multicast packet forwarding, not IP unicast packet forwarding. IP static multicast route is stored locally rather than be advertised or forwarded.

Use the following command to configure IP multicast static routes.

Command	**Function**
Ruijie (config) # **ip mroute** [**vrf** *vrf-name*] *source-address mask* { **fallback-lookup** { **global** | **vrf** *vrf-name* } | [**bgp** | **isis** | **ospf** | **static**] {*v4rpf-address* | *interface-type interface-number* } } [*distance*]	Configures IP multicast static route. The routing protocol type can be set. *distance*: In the range of 0 to 255

Caution

To set the egress of the static multicast route not to be the next hop IP address, the egress must be a point-to-point interface.

Configuring Longest-match-based Routing

Use the following command to configure longest-match-based routing.

Command	**Function**
Ruijie(config)# **ip multicast** [**vrf** *vrf-name*] **rpf longest-match**	Selects an optimal route from the static multicast, MBGP and unicast routing tables respectively in compliance with the RPF rules. Select the one with the longest mask matching from the three routes as the RPF route. If the three are of the same priority, select one in the order of multicast static route, MBGP route and unicast route.

The static multicast route, MBGP route and unicast route used for RPF check are elected from the static multicast routing table, MBGP routing table and unicast routing by RPF rules, respectively.

- By default, the one of highest priority is selected from these three routes. If they are of the same priority, select one in the order of static multicast route, MBGP route and unicast route.
- If the route selection based on the longest match has been configured, the one with the longest mask matching is selected from the three routes as the RPF route. If the masks of the three routes are the same, the one of the highest priority will be selected; if the three are of the same priority, a route is selected in the order of multicast static route, MBGP route and unicast route.

Configuring the Selection Method for PROXY in RPF Vector

Use the following commands to configure the selection method for proxy in RPF vector in global configuration mode.

Command	Function
ip multicast [**vrf** *vrf-name*] **rpf proxy** [**rd**] {**vector |** **disable**}	Configures the selection method for proxy. There are three selection methods for proxy: When the **rd** parameter is configured, the RPF Vector with RD information carried is used. This parameter takes effect only when the vrf is specified. When the **vector** parameter is configured, RFP Vector is used. When the **disable** parameter is configured, receiving RPF Vector is prohibited.
no ip multicast [**vrf** *vrf-name*] **rpf proxy** [**rd**] {**vector |** **disable**}	Deletes the selection method for proxy.

☑ Currently, this function is supported by RSR20. RSR30, RSR50 and RSR50E.

Configuring the Multicast Hardware Table Overflow Override Mechanism

This command deletes the hardware forwarding entry created earliest if the number of hardware forwarding table is full during creation of a new entry.

Command	Function
Ruijie(config)# **msf ipmc-overflow override**	Deletes the hardware forwarding entry created earliest and adds the new entry if the hardware forwarding table is full during creation of a new entry.

Monitoring and Maintaining IP Multicast Routing

Use the following command to show the IPv4 multicast forwarding table in privileged EXEC mode.

Command	Function
show ip mroute [**vrf** *vrf-name*] [*group-or-source-address* [*group-or-source-address*]] [**dense** | **sparse**] [**summary** | **count**]	Shows the IPv4 multicast forwarding table.

Use the following command to delete the IPv4 multicast forwarding table in privileged EXEC mode.

Command	Function
clear ip mroute [vrf *vrf-name***]** {*** |** *v4group-address* [*v4source -address*]}	Deletes the IPv4 multicast forwarding table.

Use the following command to reset the IPv4 multicast forwarding table statistics in privileged EXEC mode.

Command	Function
clear ip mroute [vrf *vrf-name***] statistics** {* | *v4group-address* [*v4source-address*]}	Resets the IPv4 multicast forwarding table statistics.

Use the following command to show the IPv4 static multicast routing information in privileged EXEC mode.

Command	Function
show ip mroute [**vrf** *vrf-name*] **static**	Shows the IPv4 static multicast routing information.

Use the following command to show the IPv4 static multicast routing information in privileged EXEC mode.

Command	Function
show ip rpf [**vrf** *vrf-name*] { *source-address* [*group-address*] [**rd** *route-distinguisher*] } [**metric**]	Shows the RPF information of specific IPv4 source address.

☑ Only RSR20, RSR30, RSR50 and RSR50E support the parameters of **group address**, **rd** and **metric**.

Use the following command to show the IPv4 static multicast interface information in privileged EXEC mode.

Command	Function
show ip mvif [**vrf** *vrf-name*] [*interface-type interface-number*]	Shows the IPv4 multicast interface information.

Use the following command to show the IPv4 layer 3 multicast forwarding table in privileged EXEC mode.

Command	Function
show ip mrf [**vrf** *vrf-name*] **mfc**	Shows the IPv4 layer 3 multicast forwarding table.

Use the following command to show the operation of multicast core in privileged mode.

Command	Function
debug nsm mcast [**vrf** *vrf-name*] **all**	Shows the operation of multicast core.

Use the following command to show the communication between the core of IPv4 multicast and multicast protocols in privileged mode.

Command	Function
debug nsm mcast [**vrf** *vrf-name*] **fib-msg**	Shows the communication between the core of IPv4 multicast and multicast protocols.

Use the following command to show the operation on the interface of the core of IPv4 multicast in privileged mode.

Command	Function
debug nsm mcast [**vrf** *vrf-name*] **vif**	Shows the operation on the interface of the core of IPv4 multicast.

Use the following command to show the operation of interface and statistics of the core of IPv4 multicast in privileged mode.

Command	Function
debug nsm mcast [**vrf** *vrf-name*] **stats**	Shows the operation of interface and statistics of the core of IPv4 multicast.

Use the following command to show the packet forwarding on Layer 3 of IPv4 multicast in privileged mode.

Command	Function

debug ip mrf [**vrf** *vrf-name*] **forwarding**	Shows the packet forwarding on Layer 3 of IPv4 multicast.

Use the following command to show the operation of forwarding entries on Layer 3 of IPv4 multicast in privileged mode.

Command	Function
debug ip mrf [**vrf** *vrf-name*] **mfc**	Shows the operation of forwarding entries on Layer 3 of IPv6 multicast.

Use the following command to show the operation of forwarding events on Layer 3 of IPv4 multicast in privileged mode.

Command	Function
debug ip mrf [**vrf** *vrf-name*] **event**	Shows the operation of forwarding events on Layer 3 of IPv4 multicast.

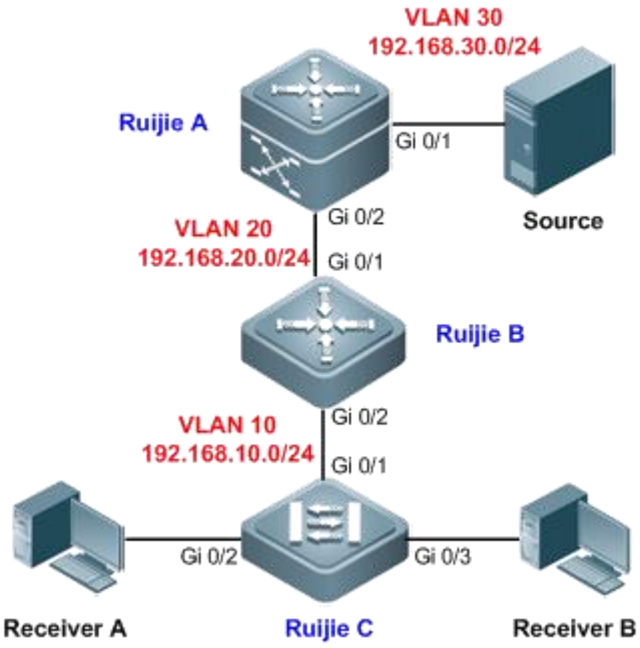
Configuration Examples

PIM-DM Configuration Example

Networking Topology

As shown in Figure 6, Ruijie A and Ruijie B are layer 3 devices; Ruijie C is a layer-2 access device, with downlink users belonging to VLAN 10. The multicast source belongs to VLAN 30, and resides in a network segment different from the multicast receivers.

Figure 6 Topology for multicast routing network



Networking Requirements

- IGMP is running between multicast source and multicast receiver to establish and maintain the membership of a multicast group. For a dense-mode multicast network, PIM-DM can be applied to realize layer-3 routing of multicast data, while IGMP Snooping can be applied on the layer-2 device to realize layer-2 forwarding of multicast data.
- Only hosts belonging to VLAN 10 can join the multicast group 225.0.0.0/8, and a host can join up to 200 multicast groups.
- PIM adjacency established between the edge multicast router (Ruijie B) and the downlink device by receiving PIM data packets should be avoided.

Configuration Tips

- Configure unicast routing protocol on the layer-3 devices (Ruijie A and Ruijie B in this example), and ensure the route connectivity between different network segments. Static route is configured in this example.
- On the layer-3 interface of multicast routing devices (SVI of VLANs 10, 20 and 30), configure PIM-DM to enable IGMP automatically (IGMPv2 is the default version).
- Configure IGMP Snooping on the layer-2 device (Ruijie C in this example). Here only the IVGL mode of IGMP Snooping is enabled, and detailed configurations are not provided here. For details, see *Configuring IGMP Snooping*.
- Configure multicast group access control on the layer-3 interface (SVI for VLAN 10 of Ruijie B) of multicast router to limit the range of multicast groups the downlink hosts can join. Configure the maximum number of IGMP group members on this interface (in this example, the maximum number is set to 200).
- Configure PIM neighbor filtering on the layer-3 interface of Ruijie B for connecting to the layer-2 device. By setting up filtering conditions for ACL, only the PIM packets from uplink neighbors can be received.

Configuration Steps

- Step 1: Configure SVI for each VLAN.

! On Ruijie A, create VLAN 20 and VLAN 30, and configure the SVI for VLAN 20 as 192.168.20.1/24 and the SVI for VLAN 30 as 192.168.30.1/24.

```
RuijieA#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
RuijieA(config)#vlan 20
RuijieA(config-vlan)#exit
RuijieA(config)#vlan 30
RuijieA(config-vlan)#exit
RuijieA(config)#interface vlan 20
RuijieA(config-if-VLAN 20)#ip address 192.168.20.1 255.255.255.0
RuijieA(config-if-VLAN 20)#exit
RuijieA(config)#interface vlan 30
RuijieA(config-if-VLAN 30)#ip address 192.168.30.1 255.255.255.0
RuijieA(config-if-VLAN 30)#exit
```

! On Ruijie B, create VLAN 10 and VLAN 20, and configure the SVI for VLAN 10 as 192.168.10.1/24 and the SVI for VLAN 20 as 192.168.20.2/24.

```
RuijieB#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
RuijieB(config)#vlan 10
RuijieB(config-vlan)#exit
RuijieB(config)#vlan 20
RuijieB(config-vlan)#exit
RuijieB(config)#interface vlan 10
RuijieB(config-if-VLAN 10)#ip address 192.168.10.1 255.255.255.0
RuijieB(config-if-VLAN 10)#exit
RuijieB(config)#interface vlan 20
RuijieB(config-if-VLAN 20)#ip address 192.168.20.2 255.255.255.0
RuijieB(config-if-VLAN 20)#exit
```

- Step 2: Configure the attributes of each port.

! On Ruijie A, configure Gi 0/1 as an access port belonging to VLAN 30 and Gi 0/2 as a trunk port.

```
RuijieA(config)#interface gigabitEthernet 0/1
RuijieA(config-if-GigabitEthernet 0/1)#switchport access vlan 30
RuijieA(config-if-GigabitEthernet 0/1)#exit
RuijieA(config)#interface gigabitEthernet 0/2
RuijieA(config-if-GigabitEthernet 0/2)#switchport mode trunk
RuijieA(config-if-GigabitEthernet 0/2)#exit
```

! On Ruijie B, configure Gi 0/1 and Gi 0/2 as trunk ports.

```
RuijieB(config)#interface range gigabitEthernet 0/1-2
RuijieB(config-if-range)#switchport mode trunk
RuijieB(config-if-range)#exit
```

! On Ruijie C, configure Gi 0/1 as a trunk port and Gi 0/2-3 as an access port belonging to VLAN 10.

```
RuijieC(config)#interface gigabitEthernet 0/1
RuijieC(config-if-GigabitEthernet 0/1)#switchport mode trunk
RuijieC(config-if-GigabitEthernet 0/1)#exit
RuijieC(config)#interface range gigabitEthernet 0/2-3
RuijieC(config-if-range)#switchport access vlan 10
RuijieC(config-if-range)#exit
```

- Step 3: Configure static route on the layer-3 device.

! On Ruijie B, configure the next-hop IP address for 192.168.30.0 as 192.168.20.1.

```
RuijieB(config)#ip route 192.168.30.0 255.255.255.0 192.168.20.1
```

! On Ruijie A, configure the next-hop IP address for 192.168.10.0 as 192.168.20.2.

```
RuijieA(config)#ip route 192.168.10.0 255.255.255.0 192.168.20.2
```

- Step 4: Enable multicast routing on the layer-3 interface.

! On Ruijie A, enable multicast routing globally, and enable PIM-DM on each interface.

```
RuijieA(config)#ip multicast-routing
RuijieA(config)#interface vlan 20
RuijieA(config-if-VLAN 20)#ip pim dense-mode
RuijieA(config-if-VLAN 20)#exit
RuijieA(config)#interface vlan 30
RuijieA(config-if-VLAN 30)#ip pim dense-mode
RuijieA(config-if-VLAN 30)#exit
```

! On Ruijie B, enable multicast routing globally, and enable PIM-DM on each interface.

```
RuijieB(config)#ip multicast-routing
RuijieB(config)#interface vlan 10
RuijieB(config-if-VLAN 10)#ip pim dense-mode
RuijieB(config-if-VLAN 10)#exit
RuijieB(config)#interface vlan 20
RuijieB(config-if-VLAN 20)#ip pim dense-mode
RuijieB(config-if-VLAN 20)#exit
```

- Step 5: Enable IGMP Snooping on the layer-2 device.

! In global configuration mode, configure IGMP Snooping to operate in IVGL mode.

```
RuijieC(config)#ip igmp snooping ivgl
```

- Step 6: Configure multicast group access control on the layer-3 interface and configure the maximum number of IGMP group members.

! On Ruijie B, create ACL to permit the IP address 225.0.0.0/8.

```
RuijieB(config)#ip access-list standard 1
RuijieB(config-std-nacl)#permit 225.0.0.0 0.255.255.255
RuijieB(config-std-nacl)#exit
```

! On the SVI for VLAN 10, configure multicast group access control and associate ACL.

```
RuijieB(config)#interface vlan 10
RuijieB(config-if-VLAN 10)#ip igmp access-group 1
```

! On the SVI for VLAN 10, configure the maximum number of allowed multicast groups as 200.

```
RuijieB(config-if-VLAN 10)#ip igmp limit 200
RuijieB(config-if-VLAN 10)#exit
```

- Step 7: Configure PIM neighbor filtering.

! On Ruijie B, create the ACL to deny all IP addresses.

```
RuijieB(config)#ip access-list standard 2
RuijieB(config-std-nacl)#deny any
RuijieB(config-std-nacl)#exit
```

! Configure PIM neighbor filtering on the SVI for VLAN 10 and associate ACL so that this interface does not receive PIM packets from other devices or establish adjacencies with them.

```
RuijieB(config)#interface vlan 10
RuijieB(config-if-VLAN 10)#ip pim neighbor-filter 2
RuijieB(config-if-VLAN 10)#exit
```

Verification

- Step 1: Display device configurations.

! Configurations on Switch A

```
RuijieA#show running-config
!
vlan 20
!
vlan 30
!
ip multicast-routing
!
interface GigabitEthernet 0/1
 switchport access vlan 30
!
interface GigabitEthernet 0/2
 switchport mode trunk
!
interface VLAN 20
 ip pim dense-mode
 no ip proxy-arp
 ip address 192.168.20.1 255.255.255.0
!
interface VLAN 30
 ip pim dense-mode
 no ip proxy-arp
 ip address 192.168.30.1 255.255.255.0
!
ip route 192.168.10.0 255.255.255.0 192.168.20.2
```

! Configurations on Switch B

```
SwtichB#show running-config
!
vlan 10
!
vlan 20
!
ip multicast-routing
!
ip access-list standard 1
 10 permit 225.0.0.0 0.255.255.255
!
ip access-list standard 2
 10 deny any
!
interface GigabitEthernet 0/1
 switchport mode trunk
!
interface GigabitEthernet 0/2
 switchport mode trunk
!
interface VLAN 10
 ip pim dense-mode
 ip pim neighbor-filter 2
 ip igmp access-group 1
 ip igmp limit 200
 no ip proxy-arp
 ip address 192.168.10.1 255.255.255.0
!
interface VLAN 20
 ip pim dense-mode
 no ip proxy-arp
 ip address 192.168.20.2 255.255.255.0
!
ip route 192.168.30.0 255.255.255.0 192.168.20.1
```

- Step 2: Display PIM-DM information of the interface (Ruijie A is used as an example).

```
RuijieA#show ip pim dense-mode interface detail
VLAN 20 (vif-id: 1):
  Address 192.168.20.1
  Hello period 30 seconds, Next Hello in 30 seconds
  Over-ride interval 2500 milli-seconds
  Propagation-delay 500 milli-seconds
  Neighbors:
    192.168.20.2
VLAN 30 (vif-id: 2):
  Address 192.168.30.1
  Hello period 30 seconds, Next Hello in 25 seconds
  Over-ride interval 2500 milli-seconds
  Propagation-delay 500 milli-seconds
  Neighbors: none
```

The preceding information shows the ID and address of the PIM-DM enabled interface and the corresponding PIM-DM neighbor.

- Step 3: Display the next hop information of PIM-DM (Ruijie B is used as an example).

```
RuijieB#show ip pim dense-mode nexthop
Destination   Nexthop   Nexthop       Nexthop       Metric Pref
              Num       Addr          Interface
192.168.30.2   1      192.168.20.1   VLAN 20           0     1
```

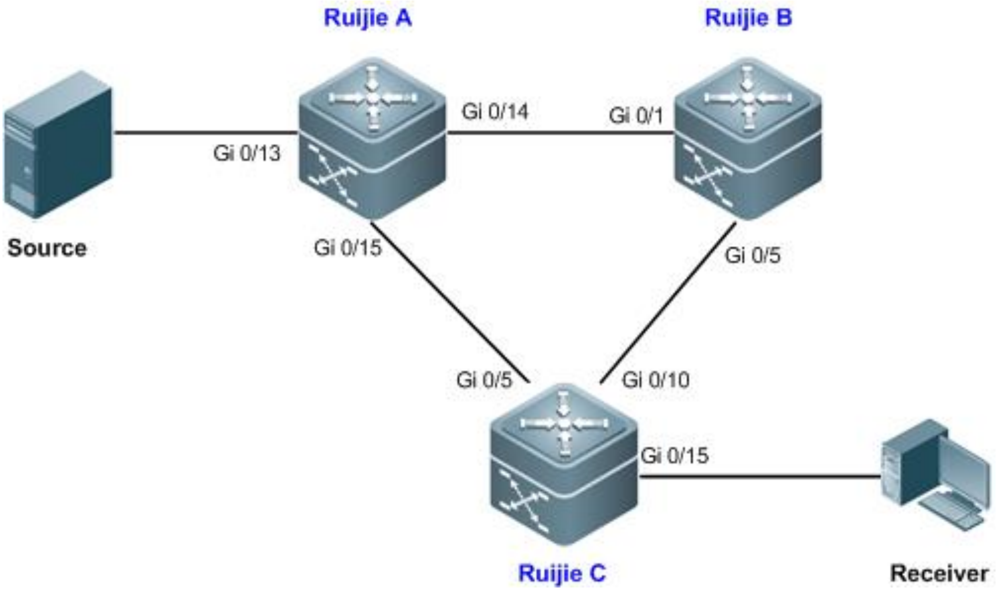
☑ This example applies to both layer-3 switches and routers. However, VLAN-related configuration commands are not supported on routers. Therefore, to establish the topology in this example on routers, directly replace the SVI interfaces with use common layer-3 interfaces.

PIM-SM Configuration Example (I)

Networking Topology

As shown in Figure 7, three layer-3 devices are interconnected through the routed ports. Multicast source and receiver are in different network segments.

Figure 7



The interface addresses of the devices are listed in the following table.

Device	Port Number	IP Address
Ruijie A	Gi 0/13	192.168.1.1/24
	Gi 0/14	192.168.2.1/24
	Gi 0/15	192.168.3.1/24
Ruijie B	Gi 0/1	192.168.2.2/24
	Gi 0/5	192.168.4.1/24
	Loopback1	10.1.1.1/24
Ruijie C	Gi 0/5	192.168.3.2/24
	Gi 0/10	192.168.4.2/24
	Gi 0/15	192.168.5.1/24

Networking Requirements

- IGMP is running between multicast source and multicast receiver to establish and maintain the membership of a multicast group. For a sparse-mode multicast network, PIM-SM can be applied to realize layer-3 routing of multicast data.
- Unauthenticated multicast source should be avoided from sending multicast data in the PIM-SM domain.

Configuration Tips

- Configure unicast routing protocols on the layer-3 devices, and ensure the route connectivity between different network segments. This example configures the OSPF protocol. For details, see *Configuring OSPF*.
- After enabling PIM-SM on each interface, IGMP will be enabled automatically (IGMPv2 is the default version).
- In the entire PIM-SM domain, at least one RP must be configured (serving all multicast groups by default) to act as the root node of shared tree (in this example, one interface of Ruijie B is configured as static RP).
- On the RP, configure address filtering of register messages (on Ruijie B in this example).

Configuration Steps

- Step 1: Configure the IP address on the interface of each device.

! Configure the IP address of Ruijie A's interface.

```
RuijieA#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
RuijieA(config)#interface gigabitEthernet 0/13
RuijieA(config-if-GigabitEthernet 0/13)#no switchport
RuijieA(config-if-GigabitEthernet 0/13)#ip address 192.168.1.1 255.255.255.0
RuijieA(config-if-GigabitEthernet 0/13)#exit
RuijieA(config)#interface gigabitEthernet 0/14
RuijieA(config-if-GigabitEthernet 0/14)#no switchport
RuijieA(config-if-GigabitEthernet 0/14)#ip address 192.168.2.1 255.255.255.0
RuijieA(config-if-GigabitEthernet 0/14)#exit
RuijieA(config)#interface gigabitEthernet 0/15
RuijieA(config-if-GigabitEthernet 0/15)#no switchport
RuijieA(config-if-GigabitEthernet 0/15)#ip address 192.168.3.1 255.255.255.0
RuijieA(config-if-GigabitEthernet 0/15)#exit
```

! Configure the IP address of Ruijie B's interface, and configure a Loopback interface as well.

```
RuijieB(config)#interface gigabitEthernet 0/1
RuijieB(config-if-GigabitEthernet 0/1)#no switchport
RuijieB(config-if-GigabitEthernet 0/1)#ip address 192.168.2.2 255.255.255.0
RuijieB(config-if-GigabitEthernet 0/1)#exit
RuijieB(config)#interface gigabitEthernet 0/5
RuijieB(config-if-GigabitEthernet 0/5)#no switchport
RuijieB(config-if-GigabitEthernet 0/5)#ip address 192.168.4.1 255.255.255.0
RuijieB(config-if-GigabitEthernet 0/5)#exit
RuijieB(config)#interface loopback 1
RuijieB(config-if-Loopback 1)#ip address 10.1.1.1 255.255.255.0
RuijieB(config-if-Loopback 1)#exit
```

! Configure the IP address of Ruijie C's interface.

```
RuijieC(config)#interface gigabitEthernet 0/5
RuijieC(config-GigabitEthernet 0/5)#no switchport
RuijieC(config-GigabitEthernet 0/5)#ip address 192.168.3.2 255.255.255.0
RuijieC(config-GigabitEthernet 0/5)#exit
RuijieC(config)#interface gigabitEthernet 0/10
RuijieC(config-GigabitEthernet 0/10)#no switchport
RuijieC(config-GigabitEthernet 0/10)#ip address 192.168.4.2 255.255.255.0
RuijieC(config-GigabitEthernet 0/10)#exit
RuijieC(config)#interface gigabitEthernet 0/15
RuijieC(config-GigabitEthernet 0/15)#no switchport
RuijieC(config-GigabitEthernet 0/15)#ip address 192.168.5.1 255.255.255.0
```

- Step 2: Interconnect devices and configure the corresponding OSPF protocol on the devices.

! Configure Switch A

```
RuijieA(config)#route ospf 1
RuijieA(config-router)#network 192.168.1.0 0.0.0.255 area 0
RuijieA(config-router)#network 192.168.2.0 0.0.0.255 area 0
RuijieA(config-router)#network 192.168.3.0 0.0.0.255 area 0
RuijieA(config-router)#exit
```

! Configure Ruijie B

```
RuijieB(config)#route ospf 1
RuijieB(config-router)#network 10.1.1.0 0.0.0.255 area 0
RuijieB(config-router)#network 192.168.2.0 0.0.0.255 area 0
RuijieB(config-router)#network 192.168.4.0 0.0.0.255 area 0
RuijieB(config-router)#exit
```

! Configure Ruijie C

```
RuijieC(config)#route ospf 1
RuijieC(config-router)#network 192.168.3.0 0.0.0.255 area 0
RuijieC(config-router)#network 192.168.4.0 0.0.0.255 area 0
RuijieC(config-router)#network 192.168.5.0 0.0.0.255 area 0
RuijieC(config-router)#exit
```

- Step 3: Globally enable multicast routing on the devices and enable PIM-SM on each interface.

! Configure Ruijie A

```
RuijieA(config)#ip multicast-routing
RuijieA(config)#interface gigabitEthernet 0/13
RuijieA(config-if-GigabitEthernet 0/13)#ip pim sparse-mode
RuijieA(config-if-GigabitEthernet 0/13)#exit
RuijieA(config)#interface gigabitEthernet 0/14
RuijieA(config-if-GigabitEthernet 0/14)#ip pim sparse-mode
RuijieA(config-if-GigabitEthernet 0/14)#exit
RuijieA(config)#interface gigabitEthernet 0/15
RuijieA(config-if-GigabitEthernet 0/15)#ip pim sparse-mode
RuijieA(config-if-GigabitEthernet 0/15)#exit
```

! Configurations on Ruijie B and Ruijie C (including the Loopback interface on Ruijie B) are the same as the preceding configurations.

- Step 4: Configure RP.

! Select the Loopback interface of Switch B as the static RP of PIM-SM domain. Note: Static RP must be configured identically on all PIM devices.

```
RuijieA(config)#ip pim rp-address 10.1.1.1
```

! Configurations on Ruijie B and Ruijie C are the same as the preceding configuration.

- Step 5: Configure address filtering of register messages on RP.

! On Ruijie B, create ACL to permit register messages with source IP address being 192.168.1.2 and group address range being 225.0.0.0/8–226.0.0.0/8.

```
RuijieB(config)#ip access-list extended 100
RuijieB(config-ext-nacl)#permit ip host 192.168.1.2 225.0.0.0 0.255.255.255
RuijieB(config-ext-nacl)#permit ip host 192.168.1.2 226.0.0.0 0.255.255.255
RuijieB(config-ext-nacl)#deny ip any any
RuijieB(config-ext-nacl)#exit
```

! Associate this ACL with the register message address filtering of RP.

```
RuijieB(config)#ip pim accept-register list 100
```

Verification

- Step 1: Display device configurations.

! Configurations on Ruijie A.

```
RuijieA#show running-config
!
ip pim rp-address 10.1.1.1
!
ip multicast-routing
!
interface GigabitEthernet 0/13
 no switchport
 ip pim sparse-mode
 no ip proxy-arp
 ip address 192.168.1.1 255.255.255.0
!
interface GigabitEthernet 0/14
 no switchport
 ip pim sparse-mode
 no ip proxy-arp
 ip address 192.168.2.1 255.255.255.0
!
interface GigabitEthernet 0/15
 no switchport
 ip pim sparse-mode
 no ip proxy-arp
 ip address 192.168.3.1 255.255.255.0
!
router ospf 1
 network 192.168.1.0 0.0.0.255 area 0
 network 192.168.2.0 0.0.0.255 area 0
 network 192.168.3.0 0.0.0.255 area 0
!
```

! Configurations on Ruijie B.

```
RuijieB#show running-config
!
ip pim rp-address 10.1.1.1
ip pim accept-register list 100
!
```

```
ip multicast-routing
!
ip access-list extended 100
 10 permit ip host 192.168.1.2 225.0.0.0 0.255.255.255
 20 permit ip host 192.168.1.2 226.0.0.0 0.255.255.255
 30 deny ip any any
!
interface GigabitEthernet 0/1
 no switchport
 ip pim sparse-mode
 no ip proxy-arp
 ip address 192.168.2.2 255.255.255.0
!
interface GigabitEthernet 0/5
 no switchport
 ip pim sparse-mode
 no ip proxy-arp
 ip address 192.168.4.1 255.255.255.0
!
interface Loopback 1
 ip pim sparse-mode
 ip address 10.1.1.1 255.255.255.0
!
router ospf 1
 network 10.1.1.0 0.0.0.255 area 0
 network 192.168.2.0 0.0.0.255 area 0
 network 192.168.4.0 0.0.0.255 area 0
```

! Configurations on Ruijie C.

```
RuijieC#show running-config
!
ip pim rp-address 10.1.1.1
!
ip multicast-routing
!
interface GigabitEthernet 0/5
 no switchport
 ip pim sparse-mode
 no ip proxy-arp
 ip address 192.168.3.2 255.255.255.0
!
interface GigabitEthernet 0/10
 no switchport
 ip pim sparse-mode
 no ip proxy-arp
 ip address 192.168.4.2 255.255.255.0
!
interface GigabitEthernet 0/15
 no switchport
 ip pim sparse-mode
 no ip proxy-arp
 ip address 192.168.5.1 255.255.255.0
!
router ospf 1
 network 192.168.3.0 0.0.0.255 area 0
 network 192.168.4.0 0.0.0.255 area 0
 network 192.168.5.0 0.0.0.255 area 0
```

- Step 2: Display PIM-SM interface information (Ruijie B is used as an example).

```
RuijieB#show ip pim sparse-mode interface detail
GigabitEthernet 0/1 (vif 1):
  Address 192.168.2.2, DR 192.168.2.2
  Hello period 30 seconds, Next Hello in 1 seconds
  Triggered Hello period 5 seconds
  Neighbors:
   192.168.2.1
GigabitEthernet 0/5 (vif 2):
  Address 192.168.4.1, DR 192.168.4.2
  Hello period 30 seconds, Next Hello in 10 seconds
  Triggered Hello period 5 seconds
  Neighbors:
   192.168.4.2
Loopback 1 (vif 3):
  Address 10.1.1.1, DR 10.1.1.1
  Hello period 30 seconds
  Triggered Hello period 5 seconds
  Neighbors:
```

The preceding information shows the IP address of each interface and the IP addresses of the DR and PIM-SM neighbor in the corresponding network segment.

- Step 3: Display current RP information (Ruijie B is used as an example).

```
RuijieB#show ip pim sparse-mode rp mapping
PIM Group-to-RP Mappings
Group(s): 224.0.0.0/4, Static
   RP: 10.1.1.1 , Static
        Uptime: 01:43:07
```

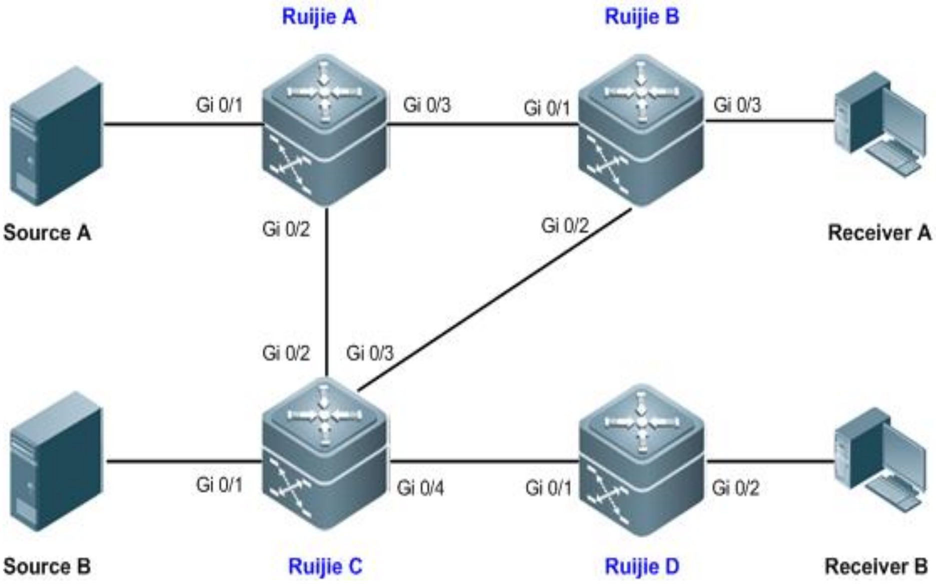
☑ This example applies to both layer-3 switches and routers. However, if you need to configure the switch port as a layer-3 port, you must run the **no switchport** command (not needed for routers).

PIM-SM Configuration Example (II)

Networking Topology

As shown in Figure 8, four layer-3 devices are interconnected through the routed ports. Multicast sources (Source A and Source B) and receivers (Receiver A and Receiver B) are in different network segments.

Figure 8



The interface addresses of each device are listed in the following table.

Device	Port Number	IP Address
Ruijie A	Gi 0/1	192.168.1.1/24
	Gi 0/2	192.168.2.1/24
	Gi 0/3	192.168.3.1/24
Ruijie B	Gi 0/1	192.168.3.2/24
	Gi 0/2	192.168.4.1/24
	Gi 0/3	192.168.5.1/24
	Loopback 1	10.1.1.1/24
	Loopback 2	10.1.2.1/24
Ruijie C	Gi 0/1	192.168.6.1/24
	Gi 0/2	192.168.2.2/24
	Gi 0/3	192.168.4.2/24
	Gi 0/4	192.168.7.1/24
	Loopback 1	10.1.3.1/24
Ruijie D	Gi 0/1	192.168.7.2/24
	Gi 0/2	192.168.8.1/24

Networking Requirements

- Multicast data forwarding between multicast routers is achieved through PIM-SM. One BSR in the PIM-SM domain is responsible for collecting and advertising RP information in the domain, while multiple candidate RPs serve different multicast groups to divert network traffic.
- The multicast router receives only BSM messages from a valid BSR.
- The BSR receives only advertisement packets from valid candidate RPs.

Configuration Tips

- Enable multicast routing on all multicast routers and enable PIM-SM multicast routing protocol on the interconnected interface. Note: Enabling PIM-SM will automatically enable IGMP.
- Specify one interface (Loopback 1 on Ruijie B) as the candidate BSR and two other interfaces (Loopback 2 on Ruijie B and Loopback 1 on Ruijie C) as the candidate RPs, and configure the multicast groups served by the candidate RPs.
- On the multicast router needing to filter BSM messages, configure the range of valid BSR (in this example, enable Ruijie C to permit only the BSM messages sent from Loopback 1 of Ruijie B).
- On BSR (Ruijie B in this example), configure the elected BSR to limit the valid C-RP address range and the multicast group range served.

Configuration Steps

- Step 1: Configure the interface IP addresses of each device.

! Configure Ruijie A

```
RuijieA#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
RuijieA(config)#interface gigabitEthernet 0/1
RuijieA(config-if-GigabitEthernet 0/1)#no switchport
RuijieA(config-if-GigabitEthernet 0/1)#ip address 192.168.1.1 255.255.255.0
RuijieA(config-if-GigabitEthernet 0/1)#exit
RuijieA(config)#interface gigabitEthernet 0/2
RuijieA(config-if-GigabitEthernet 0/2)#no switchport
RuijieA(config-if-GigabitEthernet 0/2)#ip address 192.168.2.1 255.255.255.0
RuijieA(config-if-GigabitEthernet 0/2)#exit
RuijieA(config)#interface gigabitEthernet 0/3
RuijieA(config-if-GigabitEthernet 0/3)#no switchport
RuijieA(config-if-GigabitEthernet 0/3)#ip address 192.168.3.1 255.255.255.0
RuijieA(config-if-GigabitEthernet 0/3)#exit
```

! Configurations on Ruijie B, Ruijie C and Ruijie D are the same as the preceding configurations.

! On Ruijie B, configure the IP address of Loopback 1 as 10.1.1.1/24 and the IP address of Loopback 2 as 10.1.2.1/24.

```
RuijieB(config)#interface loopback 1
RuijieB(config-if-Loopback 1)#ip address 10.1.1.1 255.255.255.0
RuijieB(config-if-Loopback 1)#exit
RuijieB(config)#interface loopback 2
RuijieB(config-if-Loopback 2)#ip address 10.1.2.1 255.255.255.0
RuijieB(config-if-Loopback 2)#exit
```

! On Ruijie C, configure the IP address of Loopback 1 as 10.1.3.1/24.

```
RuijieC(config)#interface loopback 1
RuijieC(config-Loopback 1)#ip address 10.1.3.1 255.255.255.0
RuijieC(config-Loopback 1)#exit
```

- Step 2: Interconnect devices and configure the corresponding OSPF protocol on the devices.

! Configure Ruijie A

```
RuijieA(config)#route ospf 1
RuijieA(config-router)#network 192.168.1.0 0.0.0.255 area 0
RuijieA(config-router)#network 192.168.2.0 0.0.0.255 area 0
RuijieA(config-router)#network 192.168.3.0 0.0.0.255 area 0
RuijieA(config-router)#exit
```

! Configurations on Ruijie B, Ruijie C and Ruijie D are the same as the preceding configurations.

Step 2: Enable multicast routing on each device and enable PIM-SM multicast routing protocol on each interface.

! Configure Ruijie A

```
RuijieA(config)#ip multicast-routing
RuijieA(config)#interface range gigabitEthernet 0/1-3
RuijieA(config-if-range)#ip pim sparse-mode
```

! Configurations on Ruijie B, Ruijie C and Ruijie D are the same as the preceding configurations. Note: PIM-SM must be enabled on the Loopback interface.

- Step 3: Configure the candidate BSR and candidate RP.

! On Ruijie B, configure Loopback 1 as the candidate BSR.

```
RuijieB(config)#ip pim bsr-candidate loopback 1 24
```

! On Ruijie B, create standard ACL to permit the address range of 225.0.0.0/8 to 226.0.0.0/8.

```
RuijieB(config)#ip access-list standard 1
RuijieB(config-std-nacl)#permit 225.0.0.0 0.255.255.255
RuijieB(config-std-nacl)#permit 226.0.0.0 0.255.255.255
RuijieB(config-std-nacl)#exit
```

! Configure Loopback 2 of Ruijie B as the candidate RP and associate the ACL.

```
RuijieB(config)#ip pim rp-candidate loopback 2 group-list 1
```

! On Ruijie C, create the standard ACL to permit the address range of 227.0.0.0/8 to 228.0.0.0/8.

```
RuijieC(config)#ip access-list standard 1
RuijieC(config-std-nacl)#permit 227.0.0.0 0.255.255.255
RuijieC(config-std-nacl)#permit 228.0.0.0 0.255.255.255
RuijieC(config-std-nacl)#exit
```

! On Ruijie C, configure Loopback 1 as the candidate RP and associate the ACL.

```
RuijieC(config)#ip pim rp-candidate loopback 1 group-list 1
```

- Step 4: Configure the range of valid BSR.

! On Ruijie C, create the standard ACL named "bsr_acl" to permit only packets with IP address being 10.1.1.1.

```
RuijieC(config)#ip access-list standard bsr_acl
RuijieC(config-std-nacl)#permit host 10.1.1.1
RuijieC(config-std-nacl)#exit
```

! On Ruijie C, configure the range of valid BSR and associate ACL "bsr_acl".

```
RuijieC(config)#ip pim accept-bsr list bsr_acl
```

- Step 5: Configure the elected BSR to limit the valid C-RP address range and the multicast group range served.

! On Ruijie B, create extended ACL named "rp_acl" to permit only packets with IP address being 10.1.3.1 and multicast address range being 227.0.0.0/8–228.0.0.0/8.

```
RuijieB(config)#ip access-list extended rp_acl
RuijieB(config-ext-nacl)#permit ip host 10.1.3.1 227.0.0.0 0.255.255.255
RuijieB(config-ext-nacl)#permit ip host 10.1.3.1 228.0.0.0 0.255.255.255
RuijieB(config-ext-nacl)#exit
```

! On Ruijie B, configure the elected BSR to limit the valid C_RP.

```
RuijieB(config)#ip pim accept-crp list rp_acl
```

Verification

- Step 1: Display device configurations.

! Configurations on Ruijie A.

```
RuijieA#show running-config
!
ip multicast-routing
```

```
!
interface GigabitEthernet 0/1
 no switchport
 ip pim sparse-mode
 no ip proxy-arp
 ip address 192.168.1.1 255.255.255.0
!
interface GigabitEthernet 0/2
 no switchport
 ip pim sparse-mode
 no ip proxy-arp
 ip address 192.168.2.1 255.255.255.0
!
interface GigabitEthernet 0/3
 no switchport
 ip pim sparse-mode
 no ip proxy-arp
 ip address 192.168.3.1 255.255.255.0
!
router ospf 1
 network 192.168.1.0 0.0.0.255 area 0
 network 192.168.2.0 0.0.0.255 area 0

 network 192.168.3.0 0.0.0.255 area 0
```

! Configurations on Ruijie B.

```
RuijieB#show running-config
!
ip pim accept-crp list rp_acl
ip pim bsr-candidate Loopback 1 24
ip pim rp-candidate Loopback 2 group-list 1
!
ip multicast-routing
!
ip access-list standard 1
 10 permit 225.0.0.0 0.255.255.255
 20 permit 226.0.0.0 0.255.255.255
!
ip access-list extended rp_acl
 10 permit ip host 10.1.3.1 227.0.0.0 0.255.255.255
 20 permit ip host 10.1.3.1 228.0.0.0 0.255.255.255
!
interface GigabitEthernet 0/1
 no switchport
 ip pim sparse-mode
 no ip proxy-arp
 ip address 192.168.3.2 255.255.255.0
!
interface GigabitEthernet 0/2
 no switchport
 ip pim sparse-mode
 no ip proxy-arp
 ip address 192.168.4.1 255.255.255.0
!
interface GigabitEthernet 0/3
 no switchport
 ip pim sparse-mode
 no ip proxy-arp
 ip address 192.168.5.1 255.255.255.0
!
interface Loopback 1
 ip pim sparse-mode
 ip address 10.1.1.1 255.255.255.0
!
interface Loopback 2
 ip pim sparse-mode
 ip address 10.1.2.1 255.255.255.0
!
router ospf 1
 network 10.1.1.0 0.0.0.255 area 0
 network 10.1.2.0 0.0.0.255 area 0
 network 192.168.3.0 0.0.0.255 area 0
 network 192.168.4.0 0.0.0.255 area 0
 network 192.168.5.0 0.0.0.255 area 0
```

! Configurations on Ruijie C.

```
RuijieC#show running-config
!
ip pim accept-bsr list bsr_acl
ip pim rp-candidate Loopback 1 group-list 1
!
ip multicast-routing
!
ip access-list standard 1
 10 permit 227.0.0.0 0.255.255.255
 20 permit 228.0.0.0 0.255.255.255
!
ip access-list standard bsr_acl
```

```
 10 permit host 10.1.1.1
!
interface GigabitEthernet 0/1
 no switchport
 ip pim sparse-mode
 no ip proxy-arp
 ip address 192.168.6.1 255.255.255.0
!
interface GigabitEthernet 0/2
 no switchport
 ip pim sparse-mode
 no ip proxy-arp
 ip address 192.168.2.2 255.255.255.0
!
interface GigabitEthernet 0/3
 no switchport
 ip pim sparse-mode
 no ip proxy-arp
 ip address 192.168.4.2 255.255.255.0
!
interface GigabitEthernet 0/4
 no switchport
 ip pim sparse-mode
 no ip proxy-arp
 ip address 192.168.7.1 255.255.255.0
!
interface Loopback 1
 ip pim sparse-mode
 ip address 10.1.3.1 255.255.255.0
!
router ospf 1
 network 10.1.3.0 0.0.0.255 area 0
 network 192.168.2.0 0.0.0.255 area 0
 network 192.168.4.0 0.0.0.255 area 0
 network 192.168.6.0 0.0.0.255 area 0
 network 192.168.7.0 0.0.0.255 area 0
```

! Configurations on Ruijie D.

```
RuijieD#show running-config
!
ip multicast-routing
!
interface GigabitEthernet 0/2
 no switchport
 ip pim sparse-mode
 no ip proxy-arp
 ip address 192.168.8.1 255.255.255.0
!
interface GigabitEthernet 0/11
 no switchport
 ip pim sparse-mode
 no ip proxy-arp
 ip address 192.168.7.2 255.255.255.0
!
router ospf 1
 network 192.168.7.0 0.0.0.255 area 0
 network 192.168.8.0 0.0.0.255 area 0
```

- Step 2: Display RPs in the PIM-SM domain and the corresponding service multicast information (Ruijie A is used as an example).

```
RuijieA#show ip pim sparse-mode rp mapping
PIM Group-to-RP Mappings
Group(s): 225.0.0.0/8
  RP: 10.1.2.1
    Info source: 10.1.1.1, via bootstrap, priority 192
         Uptime: 01:15:16, expires: 00:02:00
Group(s): 226.0.0.0/8
  RP: 10.1.2.1
    Info source: 10.1.1.1, via bootstrap, priority 192
         Uptime: 01:15:16, expires: 00:02:00
Group(s): 227.0.0.0/8
  RP: 10.1.3.1
    Info source: 10.1.1.1, via bootstrap, priority 192
         Uptime: 01:13:30, expires: 00:02:00
Group(s): 228.0.0.0/8
  RP: 10.1.3.1
    Info source: 10.1.1.1, via bootstrap, priority 192
         Uptime: 01:13:30, expires: 00:02:00
```

By limiting the multicast address range served by each candidate RP, the multicast sources in the PIM-SM domain can be limited. The receiver cannot receive multicast data sent from a multicast source that is not within the address range served (225.0.0.0/8–228.0.0.0/8).

In addition, invalid BSMs (the source address is not 10.1.1.1) are directly dropped on the device configured with valid BSR limitation. If invalid C_RP advertisement messages are sent to BSR, after valid C_RP limitation is configured, BSR will filter invalid C_RP advertisement messages.

☑ This example applies to both layer-3 switches and routers. However, if you need to configure the switch port as a layer-3 port, you must run the **no switchport** command (not needed for routers).

Configuring IPv6 Multicast

Overview

Traditional IP transmission allows one host to transmit packets to a single host (unicast communication) or all hosts (broadcast communication). (Note that IPv6 no longer supports broadcast). Multicast, however, allows one host to send packets to some hosts (also known as group members).

The multicast application consists of the sender and the receiver. The sender can send multicast packets without needing to join a group. In contrast, the receiver can receive the multicast packets from the group only after joining the group.

Group members are dynamic. A host can join in or leave from a group at any time. Furthermore, there is no limit on the position and number of group members. A host can join in more than one group simultaneously if necessary. Consequently, the active status and the number of members of a group vary with time.

The device maintains the routing table for forwarding multicast packets by running IPv6 multicast routing protocol (for instance, PIM-SMv6) and learns the status of group members on the direct segment by running the MLDv1/v2 protocol. The device joins inan IPv6 multicast group by sending the MLD report message.

IPv6 multicast applies to one-to-many multimedia applications.

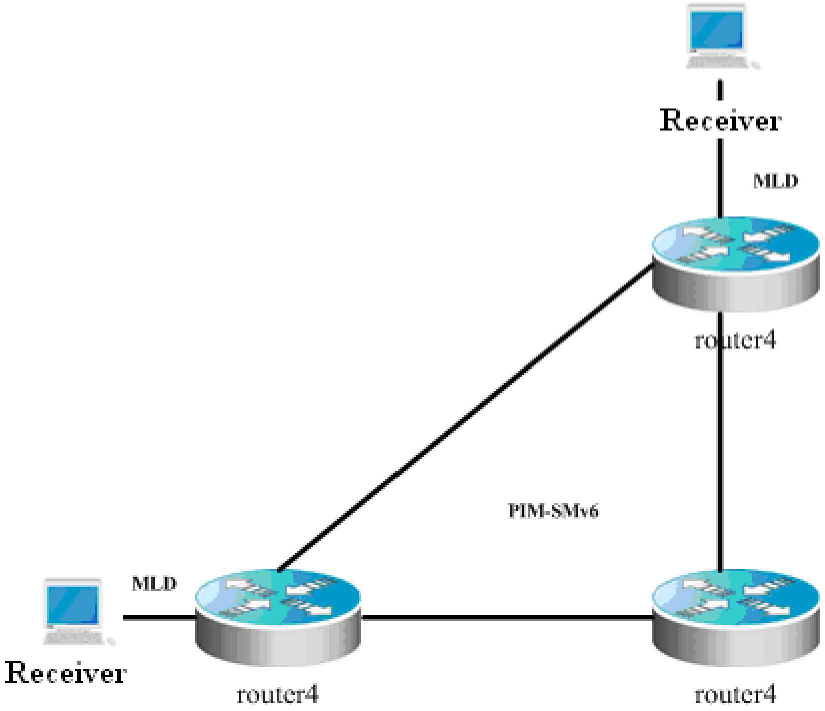
Implementation of IPv6 Multicast Routing

The IPv6 multicast routing protocol includes:

- MLD: Runs between the multicast deivce and the host to learn the relation of group members.
- PIM-SMv6: Runs between the multicast devices to enable multicast packet forwarding by setting up the multicast routing table.

The following figure illustrates the function of the multicast protocols used in IPv6 multicast packet forwarding:

Figure 1 Multicast protocols used in IPv6 multicast environment



MLD Overview

To enable IPv6 multicast, the multicast hosts and devices must support the MLD protocol, which maintains the multicast group member relation between the multicast device and the unicast device to determine the forwarding of multicast streams.

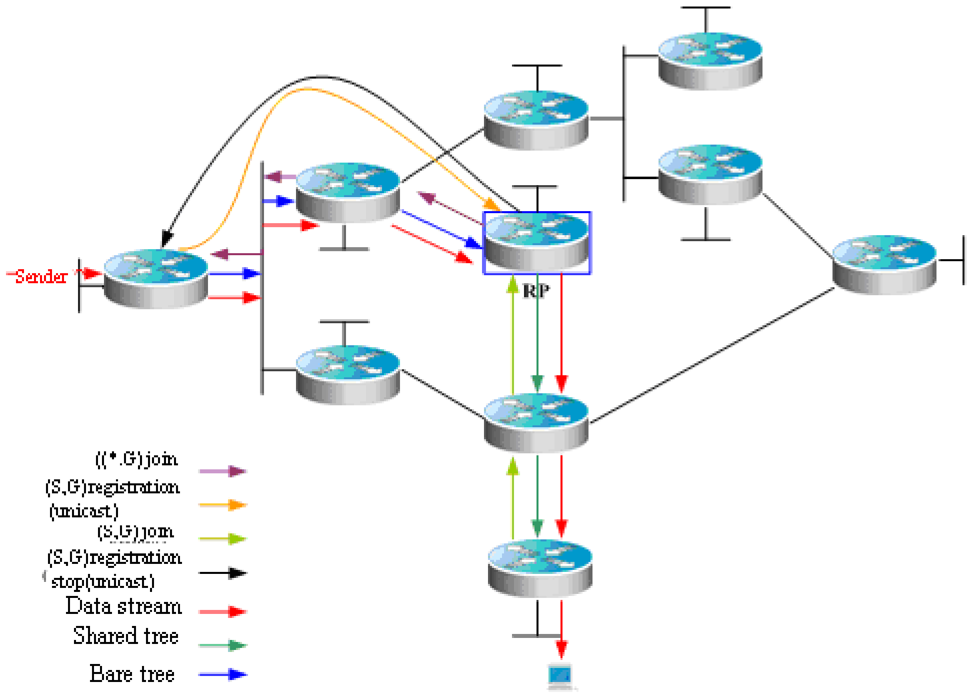
Based on the messages obtained from the MLD protocol, the device maintains a multicast group member table per interface, which is activated when at least one host on the interface is group member.

PIM-SMv6 Overview

PIM (Protocol Independent Multicast) is designed by IDMR (Inter-domain Multicast Routing) working group. As its name implied, PIM does not depend on a specific unicast routing protocol. It utilizes the unicast routing table established by various unicast routing protocols to enable the RPF check function instead of maintaining a separate multicast routing table for forwarding multicast packets. Compared with other multicast protocols, PIM overhead falls down at large extent for PIM does not need to receive and send multicast route update. The concept behind PIM design is that support and flexible transformation between SPT and the shared tree is enabled for higher multicast efficiency. There are two kinds of PIM modes-dense mode and sparse mode.

PIM-SMv6 (Protocol Independent Multicast Sparse Mode) is a multicast routing protocol in sparse mode. In the PIM-SMv6 domain, the device running PIM-SMv6 sends the Hello message at a specific interval to discover adjacent devices running PIM-SMv6 and be in charge of DR election. Here DR sends the “join/prune” message to its direct group members in the direction of the root node of the multicast distribution tree or sends the data from the direct multicast source to the multicast distribution tree.

Figure 2 PIM-SMv6 explicit join mechanism



PIM-SMv6 forwards multicast data packets by setting up a multicast distribution tree. There are two kinds of multicast distribution tree: the shared tree using G's RP as root and the shortest path tree using the multicast source as root. With the explicit join/prune mechanism, PIM-SMv6 sets up and maintains the multicast distribution tree. As shown in the above figure, when the DR receives a join report message from the receiver, it multicasts a (*.G) join message in the direction of G's RP hop by hop to join the shared tree. The source host sends multicast packets. Upon receiving the packets, its DR encapsulates them in the registration message, and then unicasts to the RP. Then the RP forwards the decapsulated packets to every group member along the shared tree. The RP sends the (S,G) join message to the first hop device along the source direction to join the shortest path tree of the source. In this way, the source sends packets along the shortest path tree to the RP without encapsulation. Upon receiving the first multicast data, the RP sends the registration stop message to the source DR to stop encapsulation. After that, the source sends multicast packets without registration and encapsulation alogn its shortest path tree to the RP, which then forwards the packets to every group members. When there is no necessary to receive multicast packets, the DR on the receiving side sends the multicast prune message to G's RP hop by hop to prune the shared tree.

RP election is also involved in PIM-SMv6. when there is one or more candidate BSRs configured in the PIM-SMv6 domain, some rule is applied to elect BSR. Candidate RPs are also configured in the PIM-SMv6 domain, which send the packets containing their addresses and serviceable multicast groups to BSRs in unicast form. BSRs generate bootstrap messages with a series of candidate RPs and the addresses of corresponding multicast groups periodically. These bootstrap messages are transmitted in the overall domain hop by hop. Devices will receive and save these bootstrap messages. Upon receiving the member relation report of a multicast group from the directly connected device, the DR will use a hash algorithm and map the multicast group address to a candidate RP who can serve this group if it has not the routing entry of this group. Then the DR will send the join/prune message in multicast form hop by hop along the way to the RP. On the other hand, upon receiving multicast packets from the directly connected device, the DR will use a hash algorithm and map the multicast group address to a candidate RP who can serve this group, and then encapsulate these multicast packets in the registration message and send it to the PR in unicast form.

The essential difference between PIM-SMv6 and PIM-DM is that PIM-SMv6 is based on explicit join mode and PIM-DM is based on flood/prune mode. For PIM-SMv6, the receiver sends a join message to the PR, but the device forwards the packets of a multicast group only on the interface joining this multicast group. PIM-SMv6 forwards multicast packets through the shared tree. Each multicast group has a rendezvous point. The multicast source sends packets to the RP along the shortest path, and then the RP sends the packets to every receiver along the shortest path. This process is similar to CBT. However, the core concept is not used in PIM-SMv6. One of the main advantages of PIM-SMv6 is that it not only receives multicast packets through the shared tree but also offers the shared tree-to-SPT transformation mechanism. This transformation consumes a lot number of resources, even though it reduces network delay and possible block on the RP. It is suitable for the environment where there are many pairs of multicast sources yet fewer networks.

PIM-SMv6 distributes multicast packets through the shared tree and SPT. Assume that other devices do not need to receive these multicast packets, unless otherwise specified. When a host joins a multicast group, the devices connecting to the host notify the root (or RP) through the PIM join message. This message is transferred among these devices in order to set up the structure of a shared tree. So, the RP records this transmission path and the registration message from the first hop device (DR) of the sending multicast source, and perfects the shared tree based on these two messages. Update of leaf messages is enabled on periodic query message. For the shared tree, the multicast source sends multicast packets to the RP so that all receivers can receive these multicast packets.

PIMv2 BSR distributes the group-to-RP message to all devices without the necessity for configuring RP for every device. The BSR distributes the mapping message through the hop-by-hop flooding BSR message. First of all, the BSR is elected among devices. This election procedure is similar to electing the root bridge in STP by priority. Every BSR device checks the BSR message, and only forwards the BSR messages with higher or equivalent priority (or higher IP address). The elected BSR sends the BSR message to the all-PIM-routers multicast group (ff02::d) with TTL 1. Upon receiving the BSR message, the adjacent PIMv2 device sents it out in multicast form and then reset TTL to 1. In this way, the BSR message is sent to all devices hop by hop. Since the BSR message includes the IP addresses of BSR devices, the candidate BSR can determine which device is the current BSR device. The candidate RP sends the candidate RP advertisement and alleges in which address ranges it can become RP. The BSR stores the advertisement message in its local candidate RP cache, and notifies all PIM devices of local candidate RPs periodically. Also in this way, the message is sent to all devices hop by hop.

RPF Rules

Multicast routing protocols depend on existing unicast route messages, MBGP routes or static multicast routes to create multicast routing entries. When creating multicast routing entries, multicast routing protocols run the Reverse Path Forwarding (RPF) check mechanism to guarantee that multicast packets are transmitted along proper paths while avoiding loops.

RPF check is on the basis of unicast routes, MBGP routes or static multicast route:

- The unicast routing table summarizes the shortest paths to each destination segments.
- The MBGP routing table directly offers multicast routes.
- The static multicast routing table lists the RPF route messages that user configures statically and manually.

The multicast routing protocol searches the unicast routing table, the MBGP routing table and the static multicast routing while run RPF check, as shown below:

1. First of all, select an optimal route from the unicast routing table, the MBGP routing table and the static multicast routing table, respectively.

a. Select an optimal route from the unicast routing table for RPF check:

- Use the IP address of the packet source as the destination address to search the unicast routing table and select an optimal unicast route.
- If the unicast route has only one next hop, check whether multicast is enabled on the egress of the next hop or not.
  - If not, the unicast route is not suitable for RPF check.
  - If so, the unicast route is suitable for RPF check and the egress serves as the RPF interface.
- If the unicast route has more than one next hop, traverse all next hops and check whether multicast is enabled on the egress of one next hop.
  - If not, traverse the next hop.
  - If so, the unicast route is suitable for RPF check and the egress serves as the RPF interface.
  - If no multicast is enabled on the egress after traversing all next hops, there is no unicast route suitable for RPF check.
- If no optical route exists, there is no unicast route suitable for RPF check.

2. Select an optimal route from the MBGP routing table for RPF check:

- Use the IP address of the packet source as the destination address to search the MBGP routing table and select an optimal MBGP route.
- The MBGP route has only one next hop. Check whether multicast is enabled on the egress of the next hop or not.
  - If not, the MBGP route is not suitable for RPF check.
  - If so, the MBGP route is suitable for RPF check.
- If no optical route exists, there is no MBGP route suitable for RPF check.

3. Select an optimal route from the static multicast routing table for RPF check:

- Use the IP address of the packet source as the destination address to search the static multicast routing table and select an optimal static multicast route.
- If the static multicast route has only one next hop, check whether multicast is enabled on the egress of the next hop or not.
  - If not, the static multicast route is not suitable for RPF check.
  - If so, further check whether there is a unicast route available for RPF check.
    - If the next hop does not associate with the unicast protocol number, the static multicast route is suitable for RPF check.
    - If there is no unicast route available for RPF check, the static multicast route is suitable for RPF check.
    - If there is a unicast route available for RPF check, but the unicast protocol number is inconsistent with the one associated with the next hop of the static multicast route, the static multicast route is not suitable for RPF check.

    - ➢ If there is a unicast route available for RPF check, and the unicast protocol number is consistent with the one associated with the next hop of the static multicast route, the static multicast route is suitable for RPF check.

- ◆ If the static multicast route has more than one next hop, traverse all next hops and check whether multicast is enabled on the egress of one next hop.

  - ■ If not, traverse the next hop.
  - ■ If so, further check whether there is a unicast route available for RPF check.
    - ➢ If the next hop does not associate with the unicast protocol number, the static multicast route is suitable for RPF check.
    - ➢ If there is no unicast route available for RPF check, the static multicast route is suitable for RPF check.
    - ➢ If there is a unicast route available for RPF check, but the unicast protocol number is inconsistent with the one associated with the next hop of the static multicast route, traverse the next hop.
    - ➢ If there is a unicast route available for RPF check, and the unicast protocol number is consistent with the one associated with the next hop of the static multicast route, the static multicast route is suitable for RPF check.
    - ➢ If no multicast is enabled on the egress after traversing all next hops, there is no static multicast route suitable for RPF check.

- ◆ If no optical route exists, there is no static multicast route suitable for RPF check.

4. Then, select one from these three optimal routes for RPF check.

   a. If the longest match routing rule is configured, select the longest match route; if these three routes are of the same mask, select the one of the highest priority; if they are of the same priority, select the one in the order of static multicast route, MBGP route and unicast route.

   b. If the longest match routing rule is not configured, select the one of the highest priority; if they are of the same priority, select the one in the order of static multicast route, MBGP route and unicast route.

Caution

The effectiveness of MBGP routes recurs on unicast routes rather than distance. The implementation of current MBGP protocols does not support equal-cost routes.

Caution

The effectiveness of static multicast routes recurs on unicast routes rather than distance.

Caution

If the static unicast route is selected as RPF route, the route must be configured with the next hop IP address. PIM protocol will select RPF neighbors based on this next hop IP address. If the static unicast route is not configured with the next hop IP address, PIM protocol can get RPF neighbors.

Basic IPv6 Multicast Route Configuration

Basic IPv6 multicast route configuration includes:

- Enable IPv6 multicast route forwarding (mandatory)
- Enable IPv6 multicast route protocol (mandatory)

Enabling IPv6 Multicast Route Forwarding

This function enables software to forward multicast packets.

To enable IPv6 multicast route forwarding, run the following command in the global configuration mode:

Command	Function
Ruijie(config)# **ipv6 multicast-routing**	Enable IPv6 multicast routing.

Enabling IPv6 Multicast Route Protocol

To enable IPv6 PIM-SM multicast route protocol, run the following command in the interface configuration mode:

Command	Function
Ruijie(config-if)# **ipv6 pim sparse-mode**	Enable PIM-SM. This command should run on Layer 3 interfaces.

The following example configures PIM-SM on GigabitEthernet 0/3.

```
Ruijie(config)# ipv6 multicast-routing
Ruijie(config)# interface  gigabitEthernet 0/3
Ruijie(config-if)# ipv6 address 3333::3333/64
Ruijie(config-if)# ipv6 pim sparse-mode
```

Note

When IPv6 multicast route is enabled globally, enabling PIM-SMv6 on an interface will enable MLD on the corresponding interface. Only one kind of multicast routing protocol is allowed to run on an interface.

Caution

After enabling the layer3 multicasting on the Private VLAN and Super VLAN, if the multicast source exists in the Sub-VLAN, one more route entry is needed to be duplicated and the ingress is the Sub-VLAN in which the multicast streams enter as the ingress validity check is required when multicast forwarding, resulting in occupying one more multicast hardware entry with 1 less multicast capacity.

Caution

S5760 supports running RPF check on SVI interface, not on routed interface. Consequently, if the routed interface becomes RPF interface, multicast streams from a non-RPF interface can also be forwarded according to route entries.

Enabling MLD

Enabling IPv6 multicast route forwarding and IPv6 multicast route protocol will enable MLD.

Advanced IPv6 Multicast Core Function Configuration

Advanced IPv6 multicast core function configuration includes:

- Limit the number of the routes that are allowed to join the IPv6 multicast routing table (optional)
- Set the IPv6 multicast border for the specific IPv6 group range (optional)
- Configure static IPv6 multicast route (optional)
- IPV6 multicast route monitoring and maintenance (optional)

Limiting the Number of the Routes That are Allowed to Join the IPv6 Multicast Routing Table

In the global configuration mode, use the **ipv6 multicast route-limit** *limit* [*threshold*] command to limit the number of the routes that are allowed to join the multicast routing table. Use the no form of this command to restore it to the default value, or 1024.

Command	Function
Ruijie(config-if)# **ipv6 multicast route-limit** *limit* [*threshold*]	Limits the number of the routes that are allowed to join the multicast routing table. *limit*: Number of the routes that are allowed to join the multicast routing table in the range 1 to 2147483647, 1024 by default. Threshold: (optional) Number of multicast routes triggering alarm, 2147483647 by default. Note: Given the hardware resource for different models of devices, the routes exceeding the hardware entry threshold need to be forwarded through software, and resulting in decrease in performance.

Setting IPv6 Multicast Border for Specific IPv6 Group Range

In interface configuration mode, use the **ipv6 multicast boundary** *access-list-name* command to set IPv6 multicast border for specific IPv6 group range. Use no form of this command to restore it to the default value, namely no multicast border.

Command	Function
Ruijie(config-if)# **ipv6 multicast boundary** *access-list-name* [**in | out**]	Set IPv6 multicast border for specific IPv6 group range. ACL can be used to specify the IPv6 group range. Note: The ACL associated with this command supports only matching destination IP address, not multicast IP address and source IP address.

This command filters MLD and PIM-SMv6 protocol packets correlating with the specific IPv6 group range. Multicast streams will income and outgoing through the multicast border interface.

Configuring Static IPv6 Multicast Route

IPv6 static multicast route enables multicast packet forwarding through a path different from IPv6 unicast path. RPF check is always performed while forwarding. The real interface receiving packets is the expected one, namely the next hop interface of IPv6 unicast route used to transmit to the sender. The check is reasonable when IPv6 unicast topology is in accord with IPv6 multicast topology. In some cases, however, it is better to make difference between IPv6 unicast path and IPv6 multicast path.

Static multicast route enables devices to execute RPF check according to configurations rather than the IPv6 unicast routing table. Consequently, tunnel technology is used for IPv6 multicast packet forwarding, not IPv6 unicast packet forwarding. IPv6 static multicast route is stored locally rather than be advertised or forwarded.

In the global configuration mode, use the following command to configure IPv6 static multicast route.

Command	Function
Ruijie(config)# **ipv6 mroute** *ipv6-prefix/prefix-length* [**bgp | isis | ospfv3 | ripng | static**]{ *ipv6-prefix* | *interface-type interface-number*} [*distance*]	Configures IPv6 static multicast route. Routing protocol can be set at the same time. Distance: <1-255>

Note

To set the egress of the static multicast route not to be the IPv6 address of next hop, the egress must be an point-to-point interface.

Configuring Longest-match-based Routing

The static multicast route, MBGP route and unicast route used for RPF check are elected from the static multicast routing table, MBGP routing table and unicast routing by RPF rules, respectively.

By default, the one of highest priority is selected from these three routes. If they are of the same priority, select one in the order of static multicast route, MBGP route and unicast route.

Use this command to select the route matching the longest mask from these three routes. If they are of the same priority level, the order of choice is: multicast static route, MBGP route and unicast route.

Command	Function
Ruijie(config)# **ipv6 multicast rpf longest-match**	Selects the route matching the longest mask.

Multicast Route Monitoring and Maintenance

In the privileged EXEC configuration mode, run the following command to show the information of IPv6 multicast forwarding table.

Command	Function
show ipv6 mroute [*v6group-address*] [*v6source-address*] [**dense**]|[**sparse**] } {[**summary**]|[**count**]}	Show the information of IPv6 multicast forwarding table.

In the privileged EXEC configuration mode, run the following command to delete the IPv6 multicast forwarding table.

Command	Function
clear ipv6 mroute {* | *v6group-address* [*v6source -address*]}	Delete the IPv6 multicast forwarding table.

In the privileged EXEC configuration mode, run the following command to reset the statics of the IPv6 multicast forwarding table.

Command	Function
clear ipv6 mroute statistics {* | *v6group-address* [*v6source-address*]}	Reset the statics of the IPv6 multicast forwarding table.

In the privileged EXEC configuration mode, run the following command to show the RPF information of the specific IPv6 source address.

Command	Function
show ipv6 rpf *v6source-address*	Show the RPF information of the specific IPv6 source address.

In the privileged EXEC configuration mode, run the following command to show the information of static IPv6 multicast route.

Command	Function
show ipv6 mroute static	Show the information of static IPv6 multicast route.

In the privileged EXEC configuration mode, run the following command to show the information of IPv6 multicast interface.

Command	Function

show ipv6 mvif [*interface-type interface-number*]	Show the information of IPv6 multicast interface.

In the privileged EXEC configuration mode, run the following command to show the IPv6 Layer 3 multicast forwarding table.

Command	Function
show ipv6 mrf mfc	Show the IPv6 Layer 3 multicast forwarding table.

In the privileged EXEC configuration mode, run the following command to show the operation of the core of IPv6 multicast.

Command	Function
debug nsm mcast6 all	Show the operation of the core of IPv6 multicast.

In the privileged EXEC configuration mode, run the following command to show the communication between the core of IPv6 multicast and multicast protocols.

Command	Function
debug nsm mcast6 fib-msg	Show the communication between the core of IPv6 multicast and multicast protocols.

In the privileged EXEC configuration mode, run the following command to show the operation on the interface of the core of IPv6 multicast.

Command	Function
debug nsm mcast6 mif	Show the operation on the interface of the core of IPv6 multicast.

In the privileged EXEC configuration mode, run the following command to show the operation of interface and statistics of the core of IPv6 multicast.

Command	Function
debug nsm mcast6 stats	Show the operation of interface and statistics of the core of IPv6 multicast.

In the privileged EXEC configuration mode, run the following command to show the packet forwarding on Layer 3 of IPv6 multicast.

Command	Function
debug ipv6 mrf forwarding	Show the packet forwarding on Layer 3 of IPv6 multicast.

In the privileged EXEC configuration mode, run the following command to show the operation of forwarding entries on Layer 3 of IPv6 multicast.

Command	Function
debug ipv6 mrf mfc	Show the operation of forwarding entries on Layer 3 of IPv6 multicast.

In the privileged EXEC configuration mode, run the following command to show the operation of forwarding events on Layer 3 of IPv6 multicast.

Command	Function
debug ipv6 mrf event	Show the operation of forwarding events on Layer 3 of IPv6 multicast.

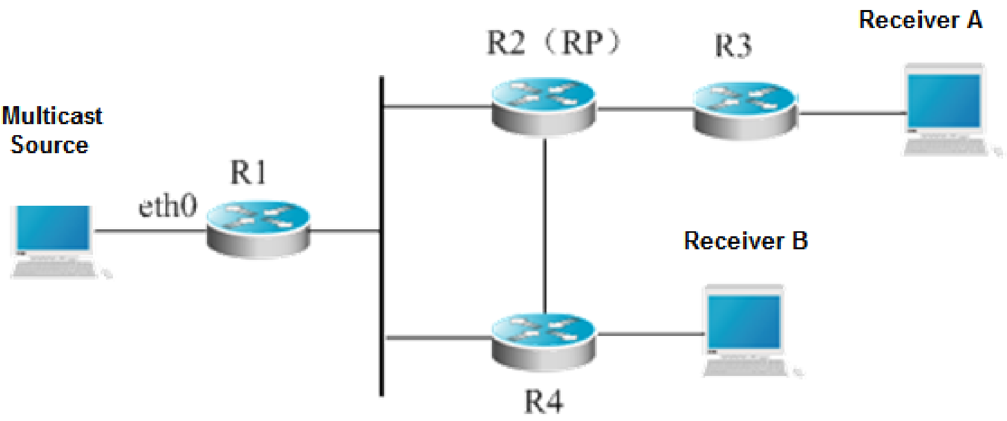
Multicast Route Configuration Example

PIM-SMv6 Configuration Example

Configuration Requirements

Figure 3 is network topology. R1 and the multicast source are located in one network. R2 is set to be RP. R3 and the Receiver A are located in the same network. R4 and the Receiver B are located in the same network. Assume that devices and hosts are connected properly, IPv6 is enabled on every interface and IPv6 unicast is enabled on every device.

Figure 3 Network topology of PIM-SMv6 configuration example



Device Configuration

Step1: Enable IPv6 multicast route.

Enable IPv6 multicast route on R1. The configurations on R2, R3 and R4 are similar.

```
Ruijie# configure terminal
Ruijie(config)# ipv6 multicast-routing
```

Step 2: Enable PIM-SMv6 on the interface.

Enable PIM-SMv6 on R1’s Gi 0/1. The configurations on R2, R3 and R4 are similar.

```
Ruijie(config)# interface GigabitEthernet 0/1
Ruijie(config-if)# ipv6 pim sparse-mode
Ruijie(config-if)# end
```

Step 3: Configure the candidate BSR and the candidate RP.

Set R2’s loopback1 to be C-BSR and C-RP

```
Ruijie(config)# interface loopback 1
Ruijie(config-if)# ipv6 address 2008:1::1/64
Ruijie(config-if)# ipv6 pim sparse-mode
Ruijie(config-if)# exit
Ruijie(config)# ipv6 pim bsr-candidate loopback 1
Ruijie(config)# ipv6 pim rp-candidate loopback 1
Ruijie(config-if)# end
```

Add the receiver into the multicast group. After the multicast source sends multicast streams, you can run **show** commands to monitor operation.

Note

When you enable PIM-SMV6, MLD automatically runs on every interface, respectively.

Configuring IGMP

IGMP Overview

IP multicast refers to a network technology that allows one or more sender (multicast source) to send one packet to more than one receiver simultaneously. The multicast source sends packets to a specific multicast group and only hosts joining the group can receive the packets. Multicast can save network bandwidth greatly because there is only a single packet transmitted on any link of the network, no matter how many receivers are deployed.

Multicast uses Class-D IP addresses specified by the Internet Assigned Numbers Authority (IANA). The four high-order bits of Class-D IP addresses are binary 1110. So, the range of multicast address is from 224.0.0.0 to 239.255.255.255. However, not all addresses in this range can be assigned to users. Some addresses are reserved for protocols or other use. For instance, the address 224.0.0.1 is assigned to all multicast hosts and 224.0.0.2 is assigned to all multicast routers.

Any hosts, no matter whether they are multicast group members or not, can be multicast sources. However, only multicast group members can receive multicast frames. A multicast group member is able to dynamically join or leave the group. Forwarding of multicast frames in the network is implemented by multicast routers running multicast routing protocols.

To enable IP multicast, hosts and routers must support the Internet Group Management Protocol (IGMP). This protocol is used by hosts to report their group memberships to multicast routers on the directly-connected network, allowing the multicast routers to determine how to forward multicast traffic. By using the information obtained from IGMP, multicast routers create an interface-based multicast group member list. The list is activated only when at least one host on an interface is a member of the group.

IGMPv1, IGMPv2 and IGMPv3 are currently supported. On the basis of IGMPv1, IGMPv2 adds a leave message for a host to actively request to leave a multicast group. IGMP behaviors includes behaviors of hosts and devices.

IGMPv1

There are only two types of IGMP messages defined in IGMPv1:

- Membership query
- Membership report

A host sends a membership report to indicate that it is interest in joining a group, and the router sends membership queries periodically to ensure that the group has at least one host. When there is no hosts in that group, the device will delete it.

IGMPv2

In IGMPv2, there are only four types of IGMP messages:

- Membership query
- Version 1 membership report
- Version 2 membership report
- Leave group

IGMPv2 is basically the same as IGMPv1, except that IGMPv2 creates a Leave group message for hosts. For IGMPv2, hosts report leave messages to routers which then send queries to check whether there is a host in the multicast group. This makes joining and leaving a group more efficient.

In the multicast network running IGMP, a multicast router is dedicated for sending IGMP query messages. This router is called a querier which is selected through an election mechanism. At first, all routers are queriers. If a router receives a query message from another router with a lower IP address, it becomes a non-querier. Consequently, there is only one querier which has the lowest IP address among all multicast routers on the network.

If a querier is invalid, new querier will be elected. . Non-queriers keep a timer for Other Querier Present Interval. Every time when a router receives a membership query packet, it resets the timer. If the timer expires, the router starts to send query messages and selects new querier again.

Queriers must periodically send membership queries to ensure that other routers on the network know that the querier still works. For this purpose, the querier maintains one query interval timer. When it sends membership query messages, this timer will be reset. When the interval timer times out, the querier sends another membership query.

When a new router appers, it sends a series of general query messages to solicit membership information. The number of general query packets depends on the Startup Query Count configured on the router. The initial general query interval is defined by the Startup Query Interval.

When a querier receives a leave group message from a host, it must send a group-specific membership query to see whether the host is the last one to leave the group. Before the querier stops forwarding packets to the group, it sends a series of such packets, the number of which is equal to the Last Member Query Count. The querier sends multiple group-specific membership queries to ensure that there is no member in the group. Such a query is sent every other the Last Member Query Interval seconds. When no response is received, the querier stops forwarding multicast packets to the group on the specified interface.

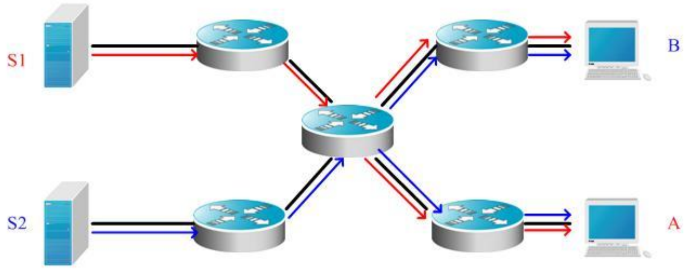
IGMPv3

Both IGMPv1 and IGMPv2 have the following defects:

- Lack of efficient measures to control multicast sources
- Difficult to establish multicast paths due to ignorance of multicast source locations.
- Difficult to find a unique multicast address. It is possible that multicast groups are using the same multicast address.

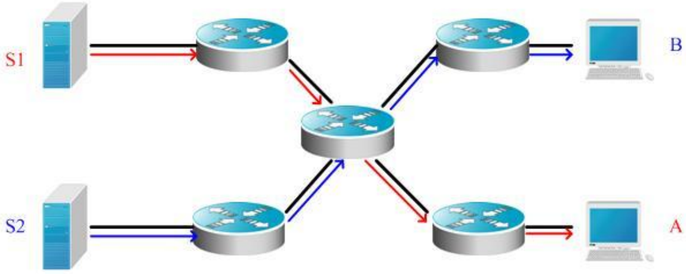
On the basis of IGMPv1 and IGMPv2, IGMPv3 provides an additional source filtering multicast function. In IGMPv1 or IGMPv2, hosts determines whether to join a group by group address and, once it joins the group, it receives multicast traffic forwarded from any source to that group address. In IGMPv3, hosts are enabled to report the multicast group they desire to join in and the multicast source from which they expect to receive traffic. A host specifies sources from which they want to receive multicast traffic through an INCLUDE list or an EXCLUDE list. Besides, IGMPv3 saves bandwidth by preventing unnecessary, illegal multicast data flows from occupying network bandwidth. It is particularly useful in the case where multiple multicast sources share one multicast address. IGMPv1 and IGMPv2 can also implement "source address filtering" in some sense, which, however, is performed on hosts receiving multicast traffic. As shown in the following diagram, two multicast sources (S1 and S2) send out traffic directed to the same multicast group address (G). This multicast traffic from S1 and S2 will be sent to all hosts receiving traffic from G. If host A only wants to receive multicast traffic from S1, it has to filter out traffic from S2 by running approriate client software.

Figure 9 Multicast traffic forwarded without source filtering



If multicast routers on the network support IGMPv3, host A wants to receive traffic from S1 only, it sends out an IGMPv3 packet in the form of “join G include S1”. host B wants to receive traffic from S2 only, it sends out an IGMPv3 packet in the form of “ join G include S2”. In this way, the traffic is forwarded as shown in Figure 2. This saves bandwidth.

Figure 10 Multicast traffic forwarded with source filtering



Based on IGMPv2, IGMPv3 adds the following two kinds of messages:

- Membership query
- Version 3 membership report

There are three types of membership query:

- General Query: used to learn information of all multicast members on an interface.
- Group-Specific Query: used to learn information of members of a specific group on an interface.
- Group-and-Source-Specific Query: a new type specified in IGMPv3 used to learn whether there is a member on an interface wants to receive group-specific multicast traffic from sources in the specified source list.

Membership Report in IGMPv3 is different from that defined in IGMPv2. The IGMPv3 membership reports are always sent with an destination address of 224.0.0.22. Besides, an IGMPv3 membership report can contain information of multiple groups.

IGMPv3 can identify membership report messages of IGMPv1 and IGMPv2 and leave group messges of IGMPv2.

IGMPv3 works the same way as IGMPv2. It is backward compatible with IGMPv1 and IGMPv2.

Caution

At most 1017 sources are allowed to forward traffic to a specific multicast group on Layer-3 interfaces of Ruijie’s switching routers. You can configure unicasting traffic from a specific allowed source. At most 1017 sources can be filtered from forwarding traffic to a specific multicast group on Layer-3 interfaces of Ruijie’s switching routers. You cannot configure unicasting traffic from the filtered sources.

IGMP Configuration Tasks

This section describes IGMP configuration tasks. Only some tasks are mandatory, and other tasks are optional depending on network requirements.

Caution All commands described in this section must be configured on layer-3 interfaces.

Default Configuration

The following table describes the default configuration of IGMP.

Feature	Default Setting
IGMP version	IGMPv2 is supported on all interfaces.
Query response interval	10 seconds
Query interval	125 seconds
Access to multicast group	All multicast groups are permitted.
Other querier present interval	255 seconds
Robustness variable	2
Last member query interval	1 second
Last member query count	2
IGMP	Disabled

Enabling IGMP

Use the following commands to enable IGMP in interface configuration mode.

Command	Function
Ruijie (config-if) # **ip pim** { **sparse-mode** | **dense-mode** }	Enables IGMP.
Ruijie (config-if) # **no ip pim** { **sparse-mode** | **dense-mode** }	Disables IGMP.

Configuring IGMP Version

Use the following commands to configure the IGMP version in interface configuration mode.

Command	Function
Ruijie (config-if) # **ip igmp version** { 1 | *2* | 3 }	Configures the IGMP version, version 2 by default.
Ruijie (config-if) # **no ip igmp version**	Restores to the default value.

Configuring Last Member Query Interval

After receiving a leave message from a multicast group, the querier sends a group-specific membership query to verify whether there is any member in the group. If no report is received during the last member query interval, the querier will regard the host that is leaving the group is the last member of that group, and then delete the information of the group. The default value of the last member query interval is 10, in units of 1/10 second. .

Use the following commands to configure the last member query interval in interface configuration mode.

Command	Function
Ruijie (config-if) # **ip igmp last-member-query-interval** *interval*	Configures the last member query interval. The *interval* specifies the range from 1 to 255. The unit is 1/10 second.
Ruijie (config-if) # **no ip igmp last-member-query-interval**	Restores to the default value.

Configuring Last Member Query Count

To prevent loss of group-specific membership query packets, it is required to send the packets for several times to ensure reliability. Therefore, you are advised to configure the last member query count to greater than 1.

Use the following commands to configure the last member query count in interface configuration mode.

Command	Function
Ruijie (config-if) # **ip igmp last-member-query-count** *count*	Configures the last member query count. The range is from 2 to 7. The default is 2.
Ruijie (config-if) # **no ip igmp last-member-query-count**	Restores to the default value.

Configuring General Query Interval

A querier sends general query messages at intervals to all hosts to verify the current membership. The destination address of the messages is the all-hosts group address, 224.0.0.1, Time To Live (TTL) is 1 and the default value is 125 seconds.

Use the following commands to configure the general query interval in interface configuration mode.

Command	Function
Ruijie (config-if) # **ip igmp query-interval** *seconds*	Configures the general query interval in seconds. The range is from 1 to 18000 seconds. The default is 125 seconds.
Ruijie (config-if) # **no ip igmp query-interval**	Restores to the default value.

Configuring the Max Response Time

The max response time is specified in the membership query message sent by the querier. Shortening this response time can allow the querier to know change of members earlier. However, it can also result in increase of the member reports diffusing in the network. Network administrators can consider a tradeoff between the two factors and then decide a proper value for the period, 10 seconds by default. Another consideration in configuring the response time is that it must be shorter than the query interval.

Use the following commands to configure the max response time in interface configuration mode.

Command	Function
Ruijie (config-if) # **ip igmp query-max-response-time** *seconds*	Configures the max response time in seconds. The range is from 1 to 25 seconds. The default value is 10 seconds.
Ruijie (config-if) # **no ip igmp query-max-response-time**	Restores to the default value.

Configuring Other Querier Present Interval

Once the timer times out, the querier considers that there is no other queriers on the network. This is helpful for the election of querier. You can reduce the value of this timer in the circumstance where the querier changes frequently to speed up response.

Use the following commands to configure the other querier present interval in interface configuration mode.

Command	Function
Ruijie (config-if) # **ip igmp query-timeout** *seconds*	Configures other querier present interval in seconds. The range is from 60 to 300 seconds. The default is 255 seconds.
Ruijie (config-if) # **no ip igmp query-timeout**	Restores to the default value.

Configuring Access Control to Multicast Groups

By default, hosts on an interface can join any multicast group. You can limit the range of multicast groups that hosts can join by configuring a standard IP ACL and applying it to the specific interface.

Use the following commands to create a standard access control list.

Command	Function
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie (config) # **access-list** *access-list-num* **permit** *A.B.C.D A.B.C.D*	Defines an ACL.
Ruijie (config)# **interface** *interface-id*	Enters interface configuration mode.
Ruijie (config-if) # **ip igmp access-group** *access-list-name*	Applies the access control list to an interface. The *access-list-name* specifies the range of group addresses that hosts on the interface can join.
Ruijie (config-if) # **no ip igmp access-group**	Deletes the access control list.

Caution

Suppose this command is associated with extended ACL. When the received IGMP report message is (S1,S2,S3…Sn,G), this command has to use an ACL entry to match the (0, G) message. Therefore you must explicitly configure an extended ACL with the (0,G) entry so that the (S1,S2,S3…Sn,G) message can be filtered.

Configuring Immediate-leave Group

In IGMPv2, you can execute this command to r reduce the leave latency of multicast group members . After this command is enabled, a host leaves a multicast group as long as it sends a leave message, without waiting the querier to send a group-specific query message. This command is available only when there is only one receiver host on an interface.

Use the following commands to configure an immdiate-leave group list. .

Command	Function
Ruijie# **config terminal**	Enters global configuration mode.

Command	Function
Ruijie (config) # **access-list** *access-list-num* **permit** *A.B.C.D A.B.C.D*	Defines an ACL.
Ruijie (config)# **interface** *interface-id*	Enters interface configuration mode.
Ruijie(config-if)# **ip igmp immediate-leave group-list** *access-list-name*	Creates an immediate-leave group list.
Ruijie (config-if) # **end**	Enters privileged EXEC mode.

Configuring Join-Group

This command configures a routeras a member of a multicast group. Use the **no** form of this command to remove the router from the multicast group.

Command	Function
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie (config)# **interface** *interface-id*	Enters interface configuration mode.
Ruijie(config-if)# **ip igmp join-group** *group-address*	Configures a router to join a multicast group.
Ruijie (config-if) # **end**	Enters privileged EXEC mode.

Use the **no ip igmp join-group** *group-address* command to remove the router from the multicast group.

Configuring Static-Group

This command configures a statically joined group. Use the **no** form of this command to remove the statically joined multicast group.

Command	Function
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie (config)# **interface** *interface-id*	Enters interface configuration mode.
Ruijie(config-if)# **ip igmp static-group** *group-address*	Configures a statically joined group.
Ruijie (config-if) # **end**	Enters privileged EXEC mode.

Use the **no ip igmp static-group** *group-address* command to remove the statically joined group. .

Configuring Limit on the Number of IGMP Group Members

Use this command to limit the number of IGMP group members globally. Membership messages that exceed the limit will not be cached or forwarded.

You can configure this command on each interface in interface or global configuration mode.

Command	Function
Ruijie(config) # **ip igmp limit** *number*	Limits the number of IGMP members in global configuration mode. The range depends on specific products.
Ruijie(config-if) # **ip igmp limit** *number*	Limits the number of IGMP members in interface

Command	Function
	configuration mode. . The range depends on specific products. By default, it is 1024.

Use the **no ip igmp limit** command to restore the default configuration.

Configuring IGMP PROXY-SERVICE

This command enables services on all the downlink mroute-proxy interfaces. After you configure this command on an interface, the interface becomes the uplink interface of the corresponding mroute-proxy service. Moreover, it associates all its downlink interfaces and maintains their propagated multicast group information.

Up to 32 proxy services can be configured using this command. The interface number with the IGMP Proxy enabled is limited by the multicast interface number supported by the device. Upon the receipt of query message, the proxy-service interface responds accordingly based on the member information that it maintains from the interfaces with mroute-proxy configured. Consequently, configuring proxy-service on an interface equals to performing host behaviors rather than router behaviors on the interface. Use the following command to configure IGMP proxy-service in interface configuration mode.

Command	Function
Ruijie(config-if)# **ip igmp proxy-service**	Configures proxy-service on the interface.

Configuring IGMP MROUTE-PROXY

This command lets an interface to forward messages to its corresponding uplink interface. The uplink interface can forward IGMP messages received from its members only when it is set to a proxy-service interface.

Use the following command to configure IGMP mroute-proxy in interface configuration mode.

Command	Function
Ruijie(config-if)# **ip igmp mroute-proxy** *interfacename*	Configures mroute-proxy on the interface. *interfacename* specifies the name of the uplink interface.

Enabling IGMP SSM-MAP

This command forcibly appends the relevant multicast source messages to the dynamically learned multicast group messages. It is usually used in conjunction with the **ip igmp ssm-map static** command.

Use the following command to enable IGMP SSM-MAP in global configuration mode.

Command	Function
Ruijie(config)# **ip igmp** [**vrf** *vrf-name*] **ssm-map enable**	Enables the SSM-MAP function under specified VRF.

Configuring IGMP SSM-MAP STATIC

This command is used in conjunction with the **ip igmp ssm-map enable** command. After this command is configured, the received messages whose version is earlier than version 3 will be mapped to the corresponding multicast source record.

Use the following command to configure IGMP SSM-MAP static in global configuration mode.

Command	Function

Command	Function
Ruijie(config)# **ip igmp** [**vrf** *vrf-name*] **ssm-map static** *access-list-num A.B.C.D*	Maps all groups matched ACL *access-list-num* under specified VRF to source address *A.B.C.D*.

Monitoring and Maintaining IGMP and Membership Information

Clearing the Dynamic Group Member Messages Obtained from the Response Message in the IGMP Cache

Use the following command to clear the dynamic group member messages obtained from the response message in the IGMP cache in privileged EXEC mode.

Command	Function
Ruijie# **clear ip igmp** [**vrf** *vrf-name*] **group**	Clears the dynamic group member messages obtained from the response message in the IGMP cache in the specified VRF.

Clearing All the Information on the Interface in IGMP Cache

Use the following command to clear all the information on the interface in IGMP cache in privileged EXEC mode.

Command	Function
Ruijie# **clear ip igmp** [**vrf** *vrf-name*] **interface** *interface-type interface-number*	Clears all the information on the interface in IGMP cache in the specified VRF.

Displaying the Status of All IGMP Group Members in the Directly-Connected Subnet

Use the following command to show the status of all IGMP group members in the directly-connected subnet in privileged EXEC mode.

Command	Function
Ruijie# **show ip igmp** [**vrf** *vrf-name*] **groups**	Shows the status of all IGMP groups under the specified VRF in the directly-connected subnet.
Ruijie# **show ip igmp** [**vrf** *vrf-name*] **groups detail**	Shows the details of all IGMP groups under the specified VRF in the directly-connected subnet.
Ruijie# **show ip igmp** [**vrf** *vrf-name*] **groups** *A.B.C.D*	Shows the status of the specified group under the specified VRF in the directly-connected subnet.
Ruijie# **show ip igmp** [**vrf** *vrf-name*] **groups** *A.B.C.D* **detail**	Shows the details of the specified group under the specified VRF in the directly-connected subnet.
Ruijie# **show ip igmp** [**vrf** *vrf-name*] **interface** *interface-type interface-number*	Shows the information of the specified interface under the specified VRF in the directly-connected subnet.
Ruijie# **show ip igmp** [**vrf** *vrf-name*] **groups** *interface-type interface-number* **detail**	Shows the details of all the groups of the specified interface under the specified VRF in the directly-connected subnet.
Ruijie# **show ip igmp** [**vrf** *vrf-name*] **groups** *interface-type interface-number A.B.C.D*	Shows the information of the specific group of the specified interface under the specified VRF in the directly-connected subnet.

Command	Function
Ruijie# **show ip igmp** [**vrf** *vrf-name*] **groups** *interface-type interface-number A.B.C.D* **detail**	Shows the details of the specific group of the specified interface under the specified VRF in the directly-connected subnet.

Displaying IGMP Interface Configuration

Use the following commands to show the IGMP interface configuration in privileged EXEC mode.

Command	Function
Ruijie# **show ip igmp** [**vrf** *vrf-name*] **interface** [*interface-type interface-number*]	Shows the configuration information of the IGMP interface in the specified VRF.
Ruijie# **show ip igmp** [**vrf** *vrf-name*] **interface**	Shows the configuration information of all the IGMP interfaces in the specified VRF.

Displaying IGMP SSM-MAP Configuration

Use the following commands to show the IGMP SSM-MAP configuration in privileged EXEC mode.

Command	Function
Ruijie# **show ip igmp** [**vrf** *vrf-name*] **ssm-mapping**	Shows the configuration information of IGMP SSM-MAP under the specified VRF.
Ruijie# **show ip igmp ssm-mapping** *A.B.C.D*	Shows the mapping information from IGMP SSM-MAP under the specified VRF to the multicast group *A.B.C.D*.

Displaying the Status of the IGMP Debugging Switch

Use the following command to show the status of the IGMP debugging switch in privileged EXEC mode.

Command	Function
Ruijie# **show debugging**	Shows the status of the IGMP debugging switch.

Turning on IGMP Debugging Switches

Use the following commands to turn on IGMP debugging switches to observe IGMP behaviors in privileged EXEC mode.

Command	Function
Ruijie# **debug ip igmp** [**vrf** *vrf-name*] **all**	Turns on all IGMP debugging switches in the specified VRF.
Ruijie# **debug ip igmp** [**vrf** *vrf-name*] **decode**	Turns on decode debugging switch in the specified VRF.
Ruijie# **debug ip igmp** [**vrf** *vrf-name*] **encode**	Turns on encode debugging switch in the specified VRF.
Ruijie# **debug ip igmp** [**vrf** *vrf-name*] **events**	Turns on event debugging switch in the specified VRF.
Ruijie# **debug ip igmp** [**vrf** *vrf-name*] **fsm**	Turns on final-state-machine debugging switch in the specified VRF.
Ruijie# **debug igmp** [**vrf** *vrf-name*] **tib**	Turns on tree debugging switch in the specified VRF.
Ruijie# **debug ip igmp** [**vrf** *vrf-name*] **warning**	Turns on warning debugging switch in the specified VRF.

Configuring MLD

MLD Overview

IPv6 multicast is a network technology allowing one or more senders (multicast sources) to send a single packet to multiple receivers (once and simultaneous). The multicast source sends the packet to a specific multicast group, and only hosts having addresses of this multicast group can receive the packet. Multicasting can substantially save network bandwidth, because only one packet is sent on any link of the entire network no matter how many destination addresses exist.

The format of IPv6 multicast address is different from that of IPv6 unicast address, and can only be used for destination address. The first byte of address format all consists of "1" (multicast address). The multicast address occupies 1/256 of the IPv6 address space. Except for parts other than the first byte, the multicast address format includes the following fields:

```
|   8    |  4 |  4 |                  112 bits                   |
 +--------+----+----+---------------------------------------------+
|11111111|flgs|scop|                  group ID
|+--------+----+----+---------------------------------------------+
```

- Flag field: (starting from high-order bit) The first bit is the reserved bit set to 0; the second bit is R flag: R = 1 indicates a multicast address that embeds the address of the RP, while R = 0 indicates a multicast address that does not embed the address of the RP; the third bit is P bit: R = 1 indicates a multicast address is created on basis of unicast prefix, otherwise it is set to 0.
- The scop field has 4 bits and indicates the scope of multicasting. According to the definition given in RFC 4291: 1 (hexadecimal number system) indicates interface-local scope, 2 indicates link-local scope, 4 indicates admin-local scope, 5 indicates site-local scope, 8 indicates organization-local scope, and E indicates global scope.
- Group ID has 112 bits, and is used to identify the multicast group. According to the multicast address (transient or well-known) and the scope of address, the same multicast ID can indicate different groups. Well-known multicast address uses specific group ID with special meanings.

No matter whether a member of multicast group or not, any host can be a multicast source. However, only the member of multicast group can receive the multicast frames. Members of the multicast group are determined dynamically, and the host can dynamically join or leave the group. The multicast frames are forwarded by the multicast device, which will run the multicast routing protocol.

To participate in IPv6 multicast, the multicast host/device shall be able to support MLD operations. This protocol allows the interactive multicasting of member relationship between host and routing device, so as to decide on the forwarding of multicast steam. By utilizing the information obtained from MLD, the device will maintain a multicast listener state table, which is based on each interface. When the local link of an interface has at least one host being a member of the group, the multicast listener state table will then be activated.

Currently, MLD has to versions. MLDv2 was developed on the basis of MLDv1 by adding the source filtering mechanism. The behaviors of MLD protocol can be divided into two parts: host behavior and router behavior.

Introduction to Messages of Different MLD versions

MLD Version 1

In MLDv1, there are three types of messages:

- Multicast Listener Query
- Multicast Listener Report
- Multicast Listener Done

In the multicast network running MLD protocol, there will be querier responsible for sending MLD query messages. Such querier is determined through election. In the beginning, all devices are of the querier state. When the devices receives multicast listener query from a device with a lower IP address, they will change from querier state to non-querier state. Therefore, only one device will be of the querier state, and this device has the lowest IP address among all multicast devices on the network.

MLDv1 also has the corresponding mechanism to handle the failure of querier device. Non-querier devices will maintain the current interval timer of other queriers. The device will reset this timer when every time it receives the multicast listener query message. If this timer times out, then the corresponding device will restart sending query message, and the new round of querier device election will begin again.

The querier device must periodically send out the multicast listener query to ensure that non-querier devices on the network know that the querier device is still workable. In realize this function, the querier device maintains a query interval timer, which will be reset when the multicast listener query message is sent out. When the query interval timer is set to zero or no longer useful, the querier device will send out another multicast listener query.

When MLD protocol is initiated, it will send a number of general query messages to discover which multicast groups shall be forwarded on the specific interface. The number of ordinary query messages sent by the device will be based on the Startup Query Count configured by the current device. The space between startup general queries will be determined by the value of Startup Query Interval. 。

When the querier device receives the leave message, it must send a group-specific multicast listener query message to see whether the host is the last listener to leave the group. The device will send a number of group-specific query messages before it stops forwarding data messages for the specific group, and the number of messages equals to the number of last listener queries. The device will send multiple group-specific multicast listener queries to ensure there is no more listener in that group. A group-specific query will be sent at every other last listener query interval in order to partition the queries. When no response is received, the device will delete the corresponding group record, and stop forwarding multicast data messages on that specific interface for this group address.

MLD Version 2

MLDV1 has the following defects in application:

- Lack of effective means to control multicast sources.
- Being unaware of the location of multicast source, it is comparatively difficult to establish the multicast route.
- It is very difficult to discover an only multicast address. Multiple multicast groups may use the same multicast address.

On the basis of MLDv1, MLDv2 provides the additional multicast source filtering mode (INCLUDE/EXCLUDE). In MLDv1, the host determines to join a certain group only according to the group address, and receives multicast streams sent from any source to this group address. In MLDv2, the host informs the host of the multicast group it desires to join in, and also notifies this host of addresses of multicast sources that it is willing/unwilling to receive. The host can indicate which sources to receive multicast streams via an INCLUDE list or an EXCLUDE list. When the host joins a multicast group:

- If the host only needs to receive the data streams sent from source { s1,s2,s3…}, then its Report message can contain the information of INCLUDE{ s1,s2,s3…}.
- If the host doesn't want to receive the data streams sent from source { s1,s2,s3…}, then its Report message can contain the information of EXCLUDE{ s1,s2,s3…}.

Through source address filtering, MLDv2 can save network bandwidth and prevent unnecessary and invalid multicast data streams from occupying the network bandwidth. This is especially useful when multiple multicast sources are sharing one multicast address. Although MLDv1 can also realize "source address filtering" in a certain sense, it is done on the receiving end of multicast streams. As shown in Fig 1, S1 and S2 are multicast sources sending data streams with the same multicast address of G. The multicast streams of S1 and S2 will be sent to all hosts receiving G. If host A only wants to receive the data streams of S1, in order to avoid the disturbance of data streams from S2, we can only use the corresponding Client software to perform filtering.

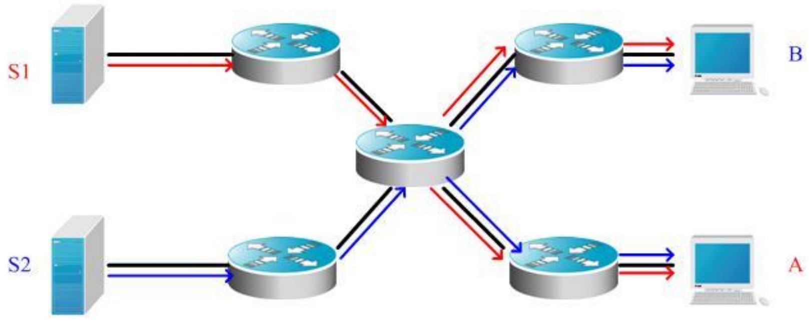


Fig 4 MLDv1 forwarding diagram

If the devices on network can support MLDv2: host A can send the MLDv2 message of join G include { S1 } if it only wants to receive data streams from S1, and host B can send the MLDv2 message of join G include { S2 } if it only wants to receive data streams from S2. The forwarding of data streams will be shown as follows, allowing the saving of partial bandwidth.

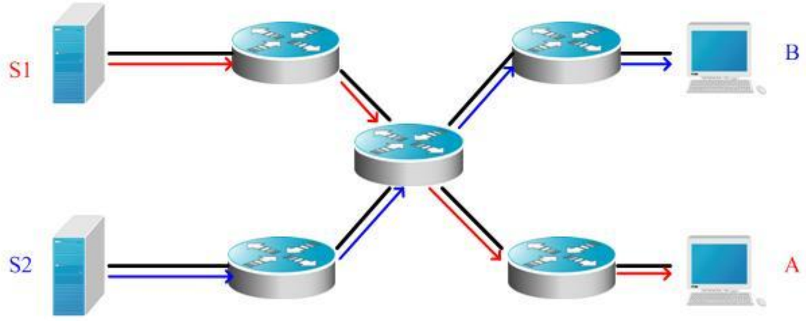


Fig 5 MLDv2 forwarding diagram

Compared with version 2, version 2 defines the following two types of messages:

- Membership Listener Query
- Membership Listener Report

Membership Query can be classified into:

- General Query: Used to learn the information of all multicast listeners on the attached link;
- Group-Specific Query: Used to learn the information of group-specific listeners on the attached link;
- Group-and-Source-Specific Query: This type is newly added by MLDv2, and is used to learn if any of listeners on the attached link needs to receive the group-specific multicast streams sent by the specified source list.

Different from the Membership Report described in MLD Version1, the Membership Report message sent by MLD Version2 may contain the information of multiple groups. The destination address of Report message is FF02::16 in MLDv2, and can carry one ore more group records, each of which contains the group address and source address. The types of group records are shown below:

- IS_IN: indicates that filtering mode between multicast group and multicast source list is INCLUDE, i.e., only the multicast data sent from the specified multicast source list to this multicast group will be received.
- IS_EX: indicates that filtering mode between multicast group and multicast source list is EXCLUDE, i.e., only the multicast data sent from multicast sources outside the specified multicast source list to this multicast group will be received.
- TO_IN: indicates that the filtering mode between multicast group and multicast source list will change from EXCLUDE mode to INCLUDE mode.
- TO_EX: indicates that the filtering mode between multicast group and multicast source list will change from INCLUDE mode to EXCLUDE mode.
- ALLOW: allows the receipt of multicast data from certain multicast sources. If the current corresponding relationship is INCLUDE, these multicast sources will be added into the existing multicast source list; if the current corresponding relationship is EXCLUDE, these multicast sources will be deleted from the existing multicast source list.
- BLOCK: no longer expects to receive multicast data from certain multicast sources. If the current corresponding relationship is INCLUDE, these multicast sources will be deleted from the existing multicast source list; if the current corresponding relationship is EXCLUDE, these multicast sources will be added into the existing multicast source list.

In consideration of compatibility, MLD Version2 can also recognize the Membership Report message and Done message of version 1.

Caution

The multicast layer-3 interface of Ruijie router can only allow up to 253 specific sources for specific groups. The user can perform the unicast of allowed specific sources.

Caution

The multicast layer-3 interface of Ruijie router can only filter up to 253 specific sources for specific groups. The user cannot perform the unicast of filtered specific sources.

MLD Protocol Specifications

The current MLD protocols include:

- RFC2710: Multicast Listener Discovery (MLD) for IPv6
- RFC3810: Multicast Listener Discovery Version 2 (MLDv2) for IPv6

MLD Configuration Task List

MLD configuration tasks include the following items, but only some configurations are compulsory, while other tasks are optional according to the specific needs of the network. Note: The following command must be configured on the layer-3 interface.

- (mandatory) Enabling MLD protocol
- (mandatory) Configuring MLD version
- (optional) Configuring Last Listener Query Interval
- (optional) Configuring Last Listener Query Count
- (optional) Configuring General Listener Query Interval
- (optional) Configuring Maximum Response Interval
- (optional) Configuring Other Querier Timer Interval
- (optional) Configuring Multicast Group Access Control
- (optional) Configuring Immediate Leave Group
- (optional) Configuring MLD Listener Number Limit
- (optional) Configure Host-Behavior Multicast Group Joining
- (optional) Configuring Static Multicast Group Joining
- (optional) Configuring MLD P
- (optional) Configuring MLD P
- (optional) Enabling MLD SSM-MAP
- (optional) Configuring MLD SSM-MAP S

Default MLD Configurations

Function	Default setting
MLD version	All interfaces support version 2
Query Response Interval	10 seconds
Query Interval	125 seconds
Multicast Group Access Control	All groups allowed
Other Querier Present Interval	255 seconds
Robustness Variable	2
Last Listener Query Interval	10 (0.1s)
Last Listener Query Count	2
MLD state	Disabled

Enabling MLD Protocol

MLD needs to be used with multicast routing protocol. If the multicast routing protocol is enabled, the MLD protocol will also be enabled. Run the following command in the interface configuration mode:

Command	Function
Ruijie (config-if) # **ipv6 pim sparse-mode**	Enable the multicast routing protocol and MLD.
Ruijie (config-if) # **no ipv6 pim sparse-mode**	Disable the multicast routing protocol and MLD.

Configuring MLD Version

To configure MLD version, run the following command in the interface configuration mode:

Command	Function
Ruijie (config-if) # **ipv6 mld version** {**1|2**}	Configure the MLD version.
Ruijie (config-if) # **no ipv6 mld version**	Recover the MLD version to the default version2.

Configuring Last Listener Query Interval

After receiving a group leave message, the querier device will send a group-specific listener query to identify that whether there are still listeners in the group. During the interval of sending last listener query, if no report is received, the device will assume that the leaving device is the last listener of the group, and will the information of this group. The default value is 1s. The Last Listener Query Interval determines the leaving speed of listeners.

Run the following command in the interface configuration mode:

Command	Function
Ruijie (config-if) # **ipv6 mld last-member-query-interval** *interval*	Configure the last listener query interval *interval*: the valid range is 1-255, in 0.1s.
Ruijie (config-if) # **no ipv6 mld last-member-query-interval**	Configure Last Listener Query Interval to default value.

Configuring Last Listener Query Count

To avoid the loss of group-specific listener query messages sent by the querier device, the messages need to be sent for several times to guarantee reliability. Therefore, the Last Listener Query Count shall be configured.

Run the following command in the interface configuration mode:

Command	Function
Ruijie(config-if) # **ipv6 mld last-member-query-count** *count*	Configure the last listener query count. *count:* the valid range is 2-7, and the default value is 2.
Ruijie(config-if) # **no ipv6 mld last-member-query-count**	Configure the last listener query count to the default value.

Configuring General Listener Query Interval

At every other listener query interval, the querier will periodically send out the listener query messages to verify the relationship of current listeners. The destination address of listener query message is all-hosts, the multicast address is FF02::1, and the TTL is 1. The default value is 125 seconds.

Run the following command in the interface configuration mode:

Command	Function
Ruijie (config-if) # **ipv6 mld query-interval** *seconds*	Configure the general listener query interval. *seconds:* the valid range is 1-18000.
Ruijie (config-if) # **no ipv6 mld query-interval**	Configure the general listener query interval to the default value of 125s.

Configuring Maximum Response Interval

This means the maximum response time required in the multicast listener query messages sent by the querier device. Less time can allow the device to rapidly learn the changes in listeners but will lead to corresponding increase in the number of potential listener reports. The network administrator can consider both factors to determine the most appropriate value, which is 10 seconds by default. Furthermore, this time shall be less than the Query Interval.

Run the following command in the interface configuration mode:

Command	Function
Ruijie (config-if)**# ipv6 mld query-max-response-time** *seconds*	Configure the maximum response time. *seconds:* the valid range is 1-25.
Ruijie (config-if)**# no ipv6 mld query-max-response-time**	Configure the maximum response time to the default value of 10s.

Configuring Other Querier Timer Interval

The configuration of other querier timer interval can help adjust the querier device election time. This value can be decreased to increase the response speed when the querier devices are prone to change.

Run the following command in the interface configuration mode:

Command	Function
Ruijie (config-if)# **ipv6 mld query-timeout** *seconds*	Configure other querier timer interval. *seconds:* the valid range is 60-300, and the default value is 255s.
Ruijie (config-if)# **no ipv6 mld query-timeout**	Configure other querier timer interval to the default value.

Configuring Multicast Group Access Control

By default, the host on one interface can join any multicast group. This feature can be used when the administrator expects to limit the scope of multicast groups which can be joined by the host. Configure an IP access list to allow and limit the scope of multicast group addresses, and apply the list to the specific interface.

Run the following commands:

Command	Function
Ruijie# **config terminal**	Enter the global configuration mode.
Ruijie (config) # **ipv6 access-list** *access-list-name* Ruijie (config-std-nacl) # **permit ipv6** source_address *group_address*	Define an access control list.
Ruijie (config)# **interface** *interface-type interface-id*	Enter the interface configuration mode.
Ruijie (config-if) # **ipv6 mld access-group** *access-list-name*	Configure the group addresses or source addresses so that multicast groups covered by the address scope of access-list-name can access this interface.
Ruijie (config-if) # **no ipv6 mld access-group**	Delete the access control list and allow the access of all groups.

Caution The Multicast Group Access Control is associated with Extended ACL. When the MLD report message received is (S1,S2,S3…Sn,G), this command will perform the corresponding ACL based check of (0,G) information. Therefore, an explicit entry of (0,G) shall be configured for extended ACL to allow the normal filtering of (S1,S2,S3…Sn,G). This (0,G) mentioned herein refers to all sources of the specific group.

Configuring Immediate Leave Group

In MLD, this command can reduce the leave latency. When there is only one host on the attached link needs to receive group information, after the host sends out the leave message, it can leave immediately without the need for the querier to device to send group-specific query. This command is used when one interface has only one receiver host.

Command	Function
Ruijie# **config terminal**	Enter the global configuration mode.
Ruijie (config)# **interface** *interface-id*	Enter the interface configuration mode.
Ruijie (config-if)# **ipv6 mld immediate-leave group-list** *access-list-name*	Configure the immediate leave group list to access-list-name.
Ruijie (config-if) # **exit**	Return to the global configuration mode.
Ruijie (config)# **ipv6 access-list** *access-list-name* Ruijie (config-std-nacl)#**permit** *x:x:x:x::x:x/< 0-128>*	Configure the address scope of group list.

Configuring MLD Listener Number Limit

This command can be used to limit the number of listeners that can be learned by MLD. Listener messages will not be learned after the listener number has exceeded the limit, and no group record will be generated.

This command can be used to configure each interface, and the interface and global can be configured separately. If exceeding the number limit of interface or global configuration, the listener messages will be dropped. Run the following commands:

Command	Function
Ruijie (config)# **ipv6 mld limit** *number*	Global configuration of MLD listener number limit Please note that different models provide different limit ranges. Please identify the default value according to the capacity indicator of different models.
Ruijie (config-if) **# ipv6 mld limit** *number*	Interface configuration of MLD listener number limit Please note that different models provide different limit ranges. The default value is 1024.
Ruijie (config-if) **# ipv6 mld limit** *number* **except** *access-list-name*	The key word of Except indicates that groups covered by ACL will be limited in number.

Use the **no ipv6 mld limit** command to restore the relevant configurations to the default values.

Configuring Host-Behavior Multicast Group Joining

Use this command to configure relevant switch interfaces to implement host behavior and join the corresponding multicast group. In this way, the sub-switches can initiatively learn the corresponding group information. This configuration can be used when a listener is required to be assigned to the interface. Use the **no** form of this command to cancel the joining operation.

Command	Function
Ruijie# **config terminal**	Enter the global configuration mode.
Ruijie (config)# **interface** *interface-id*	Enter the interface configuration mode.
Ruijie (config-if)# **ipv6 mld join-group** *group-address*	Enable the interface to join host group.
Ruijie (config-if) # **exit**	Return to the global configuration mode.

Use the **no ipv6 mld join-group** *group-address* command to leave the corresponding multicast group.

Caution

Some of the local link addresses have been reserved for use by the IP layer. If such addresses are used as the local link address, no group record will be generated.

Configuring Static Multicast Group Joining

Use this command to directly assign a listener to the relevant switch interface when a listener is required to be assigned to the interface. Use the **no** form of this command to cancel the joining operation.

Command	Function
Ruijie# **config terminal**	Enter the global configuration mode.
Ruijie (config)# **interface** *interface-id*	Enter the interface configuration mode.
Ruijie (config-if)# **ipv6 mld static-group** *group-address*	Enable the interface to join static group.
Ruijie (config-if) # **exit**	Return to the global configuration mode.

Use the **no ipv6 mld static-group** *group-address* command to delete the static group assigned to the interface.

Configuring MLD Proxy-service

Use this command to enable the service of all downlink mroute-proxy interfaces, enabling the interface to become the uplink interface of corresponding mroute-proxy. It will bind all attached downlink interfaces and maintain the group information advertised by the downlink interface.

The maximum configurable number of this command is limited to 32. The number of interfaces with MLD Proxy function enabled depends on the number of multicast interfaces supported by the device. When the interface receives the query message, the proxy-service interface will give the corresponding reply according to the listener information it maintains. Such information is acquired from the mroute-proxy interface. Therefore, proxy-service interface means that it will only implement host behavior instead of router behavior. Run the following command in the interface configuration mode:

Command	Function
Ruijie (config-if)# **ipv6 mld proxy-service**	Configure the interface to proxy-service state.

Use the **no ipv6 mld proxy-service** command to disable the proxy function on the corresponding interface.

Configuring MLD Mroute-proxy

This command will allow the interface to forward messages to the corresponding uplink interface. When the corresponding uplink interface is configured to proxy-service interface, this interface can then forward various MLD protocol messages forwarded by its listeners.

Run the following command in the interface configuration mode:

Command	Function
Ruijie (config-if ）# **ipv6 mld mroute-proxy** *interfacename*	Enables the interface to forward messages to the correspongding uplink interface.

Use the **no ipv6 mld mroute-proxy** command to cancel the binding between downlink interface and uplink interface, and disable the proxy function of downlink interface.

Enabling MLD SSM-MAP

Use this command to add the dynamically learned group information to the bound source record information, and is generally co-used with the **ipv6 mld ssm-map static** command.

Run the following command in the global configuration mode:

Command	Function
Ruijie (config)# **ipv6 mld ssm-map enable**	Enable the ssm-map function.

Use the **no ipv6 mld ssm-map enable** command to disable the SSM-MAP function.

Configuring MLD SSM-MAP Static

Use this command together with the **ipv6 mld ssm-map enable** command. After configuring this command, the MLDv1 messages received will be mapped to the corresponding source records.

Run the following command in the global configuration mode:

Command	Function
Ruijie (config ）# **ipv6 mld ssm-map static** *acl_name src_address*	Map all group records complying with acl_name to the source address of src_address.

Use the **no ipv6 mld ssm-map static** command to cancel the mapping relationship between related group record and source address.

Monitoring and Maintaining MLD State and Listener Information

Clearing the Dynamic Listener Information in the MLD Cache

Use this comamnd to clear the dynamic listener information in the MLD cache. Note that this command cannot delete the statically added listener information. Run the following command in the privilege mode:

Command	Function

Ruijie# **clear ipv6 mld group** [group-address] [interface-type interface-number]	Clear all dynamic listener information in the MLD cache learned from response message. Clear all mld group information if there is no parameter.

Clearing All Information in MLD Cache on the Specific Interface

To clear all information in MLD cache on the specific interface, run the following command in the privilege mode:

Command	Function
Ruijie# **clear ipv6 mld interface** *interface-type*	Clear all information in MLD cache on the interface.

Displaying the State of Listeners on the Attached Subnetwork

To display the state of all listeners on the attached subnetwork, run the following commands in the privilege mode:

Command	Function
Ruijie# **show ipv6 mld groups**	Display the information of all listeners learned by MLD.
Ruijie# **show ipv6 mld groups detail**	Display the detailed information of all listeners learned by MLD.
Ruijie# **show ipv6 mld groups** *x:x:x:x::x*	Display the information of group-specific listeners learned by MLD.
Ruijie# **show ipv6 mld groups** *x:x:x:x::x* detail	Display the detailed information of group-specific listeners learned by MLD.
Ruijie# **show ipv6 mld ssm-mapping**	Display the configuration information of ssm-mapping.
Ruijie# **show ipv6 mld ssm-mapping** *x:x:x:x::x*	Display the source address mapping information of specific group.
Ruijie# **show ipv6 mld groups** *interface-type interface-number* **detail**	Display the detailed information of all groups on the specific interface.
Ruijie# **show ipv6 mld groups** *interface-type interface-number x:x:x:x::x*	Display the information of specific groups on the specific interface.
Ruijie# **show ipv6 mld groups** *interface-type interface-number x:x:x:x::x* **detail**	Display the detailed information of specific groups on the specific interface.

Displaying the Configuration Information of MLD Interface

To display the configuration information of MLD interface, run the following commands in the privilege mode:

Command	Function
Ruijie# **show ipv6 mld interface** *interface-type interface-number*	Display the MLD interface configurations.
Ruijie# **show ipv6 mld interface**	Display the configurations of all MLD interfaces.

Displaying the On/Off State of MLD Debug Switch

To display the on/off state of MLD debug switch, run the following command in the privilege mode:

Command	Function
Ruijie# **show debugging**	Display the on/off state of MLD debug switch.

Turning on MLD Debug Information Switch and Observing MLD Behaviors

To turn on MLD debug information switch and observe MLD behaviors, run the following command in the privilege mode:

Command	Function
Ruijie# **debug ipv6 mld all**	Turn on all MLD debug information switches
Ruijie# **debug ipv6 mld decode**	Turn on MLD debug message decoding switch.
Ruijie# **debug ipv6 mld encode**	Turn on MLD debug message encoding switch.
Ruijie# **debug ipv6 mld events**	Turn on MLD debug event information switch.
Ruijie# **debug ipv6 mld fsm**	Turn on MLD debug state switch.
Ruijie# **debug ipv6 mld tib**	Turn on MLD debug member tree structure information switch.
Ruijie# **debug ipv6 mld warnning**	Turn on MLD debug warning switch.

Configuring PIM-DM

PIM-DM Overview

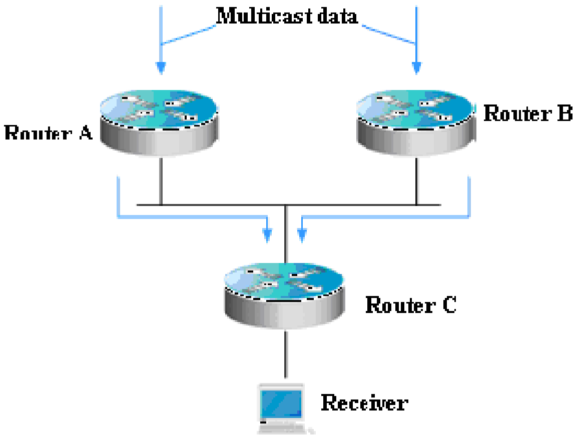
Protocol Independent Multicast-Dense Mode (PIM-DM) is a dense-mode multicast routing protocol, which is suitable for small-sized networks with densely distributed multicast members. As PIM-DM does not rely on any specific unicast routing protocol, it is called protocol independent multicast routing protocol. PIM-DM is defined in RFC 3973.

PIM-DM devices discover neighbors through Hello messages. After startup, a PIM-DM device sends a Hello message to each PIM-DM enabled interface periodically. The Hello message has a field of **Hello Hold Time**, which defines the maximum duration that a neighbor waits for the next message. If the neighbor does not receive another Hello message from the sender within this duration, this device will be removed from the adjacency list.

PIM-DM builds a shortest path tree (SPT) through flood and prune. PIM-DM assumes that when a multicast source begins to send a multicast packet, all the systems in the network need to receive this packet. As a result, this packet is forwarded to every system. The reverse path forwarding (RPF) check is performed for the packets received from the upstream interface. Those packets that fail to pass the check will be discarded. For the packets passing the check, the outgoing interface is computed based on the (S, G) pair of the packets, that is, source address and group address. If the outgoing interface is not null, an outgoing interface entry is created from the (S, G) pair and the multicast packet is forwarded through this outgoing interface. If the computed outgoing interface is null, a prune message is sent to the RPF neighbor, informing the upstream neighbor not to forward the multicast packets from the (S, G) pair to this interface. After the prune message is received on the upstream interface, the device marks the sending interface as pruned state and sets a pruned state timer. In this way, an SPT is created with the multicast source as its root.

PIM-DM uses the Assert mechanism to eliminate redundant routes.

Figure 11 Assert mechanism of PIM-DM



As shown in Figure-1, the multicast data arrives at Routers A and B at the same time, which forward the data to Router C. In this case, Router C receives duplicated data, which is not allowed. So there must be a mechanism to select Router A or B to forward the multicast data to Router C. This is the Assert mechanism of PIM-DM.

PIM-DM uses the state refresh message to update network state. The device directly connected to the multicast source sends the state refresh message to the downstream devices periodically to advertise the network topology changes. The devices receiving the message add their topology state to the state refresh message by modifying some fields, and then send it to the downstream devices. When the refresh message arrives at the leaf devices, the entire network state is updated.

PIM-DM uses the Graft mechanism to reestablish the connection with upstream devices. If the network topology of a downstream device in pruned state changes and the device needs to receive multicast data from a (S, G) pair, it sends the graft message to the upstream device. Upon receiving the graft message, the upstream device responds with a Graft-Ack message and forwards the multicast data to the downstream device again.

PIM-DM Configuration Tasks

The PIM-DM configuration tasks include the following items. However, only the first and second one are mandatory, and others are optional.

Enabling Multicast Routing

PIM-DM can forward multicast packets only after the multicast routing function is enabled.

Use the following command to enable or disable the multicast routing function in global configuration mode.

Command	Function
Ruijie (config) **# ip multicast-routing**	Enables multicast routing globally.
Ruijie (config) **# no ip multicast-routing**	Disables multicast routing globally.

Enabling PIM-DM

PIM-DM must be enabled on each interface. A device can exchange PIM-DM control messages with other devices, maintain and update the multicast routing table and forward multicast messages only after PIM-DM is enabled on the interface of the device.

Use the following commands to enable PIM-DM in interface configuration mode.

Command	Function
Ruijie(config-if)# **ip pim dense-mode**	Enables the PIM-DM protocol on the interface.
Ruijie(config-if)**# no ip pim dense-mode**	Disables the PIM-DM protocol on the interface.

Caution

Enabling PIM-DM will take effect on an interface only after the multicast routing is enabled in global configuration mode.

When this command is configured, if the system displays "Failed to enable PIM-DM on <*interface name*>, resource temporarily unavailable, please try again", configure this command again.

When this command is configured, if the system displays "PIM-DM Configure failed! VIF limit exceeded in NSM!!!", the number of configured multicast interfaces reaches the threshold. If you still need to enable PIM-DM on the interface, remove some unnecessary PIM-DM, PIM-SM or DVMRP interfaces.

It is not recommended that different IPv4 multicast routing protocols be configured on different interfaces of a switch or router.

If the interface is of tunnel-type, only 4Over4 configuration tunnel, 4Over4 GRE tunnel, 4Over6 configuration tunnel and 4Over6 GRE tunnel support the IPv4 multicasting. The multicasting function can also be enabled on other tunnel interfaces that do not support IPv4 multicasting, but no error message will be displayed and no multicast packets will be received or sent.

Multicast tunnels can be created on Ethernet interfaces only. Nested tunnel and multicast data QoS/ACL are not supported.

Setting the Interval of Sending the Hello Message

After PIM-DM is enabled on an interface, the interface sends the Hello message to the interfaces of adjacent devices periodically. You can modify the interval according to the network situation.

Use the following command to configure the interval of sending the Hello message in interface configuration mode.

Command	Function
Ruijie(config-if)# **ip pimquery-interval** *interval-seconds*	Sets the interval of sending the Hello message on the interface. *interval-seconds*: in the range of 1 to 65535 seconds
Ruijie(config-if)# **no ip pimquery-interval**	Restores the setting to the default value.

By default, the interval of sending the Hello message on the interface is 30 seconds.

Caution

When the interval of sending the Hello message is updated, Hello hold time will be updated as 3.5 times of the Hello sending interval automatically. If the interval of sending Hello message multiplying 3.5 is larger than 65535, Hello hold time is updated to 65535.

Configuring Propagation Delay of Hello Message

Options can be added to Hello messages. The default value of **propagation-delay** in **LAN Prune Delay Option** is 500 milliseconds.

Use the following command to configure the propagation delay of Hello messages in interface configuration mode.

Command	Function
Ruijie(config-if)# **ip pim propagation-delay** *interval-milliseconds*	Sets the propagation delay in the range of 1 to 32767 milliseconds.
Ruijie(config-if)# **no ip pim propagation-delay**	Restores the setting to the default value.

Configuring Override Interval of Hello Message

Options can be added to Hello messages. The default value of **override-interval** in **LAN Prune Delay Option** is 2500 milliseconds.

Use the following command to configure the override interval in interface configuration mode.

Command	Function
Ruijie(config-if)# **ip pim override-interval** *interval-milliseconds*	Sets the override interval in the range of 1 to 65535 milliseconds.
Ruijie(config-if)# **no ip pim override-interval**	Restores the setting to the default value.

Configuring PIM-DM Neighbor Filtering

The neighbor filtering function can be configured on the interface to enhance network security. With neighbor filtering enabled, PIM-DM does not establish adjacency with the neighbor or deletes the established adjacency with the neighbor as long as a neighbor is denied by the access list.

Use the following command to configure the PIM neighbor filtering function in interface configuration mode.

Command	Function
Ruijie(config-if)#**ip pim neighbor-filter** *access-list*	Enables the PIM neighbor filtering function on the current interface.
Ruijie(config-if)#**no ip pim neighbor-filter** *access-list*	Disables the PIM neighbor filtering function on the current interface.

The PIM neighbor filtering function is disabled by default on an interface.

Note

ip pim neighbor-filter command description:

Only neighbor addresses permitted by the ACL can be the PIM neighbors of the current interface.

Configuring PIM-DM State Refresh

After PIM-DM is enabled on a device, if the RPF interface is directly connects to the multicast source (that is, the PIM interface is in the same network segment as the multicast source), the device periodically sends state refresh messages to downstream devices to update the entire network state. You can disable processing or forwarding of PIM state refresh messages in global configuration mode.

Use the following command to configure PIM-DM state refresh in global configuration mode.

Command	Function
Ruijie(config-if)#**no ip pim state-refresh disable**	Enables processing or forwarding PIM-DM state refresh messages.
Ruijie(config-if)# **ip pim state-refresh disable**	Disables processing or forwarding PIM-DM state refresh message.

The PIM-DM state refresh function is enabled by default.

Caution

Disabling the state refresh messages may cause the re-convergence of the converged PIM-DM multicast forward tree, resulting in unnecessary bandwidth waste and routing table flapping. Therefore, it is better not to disable the state refresh function in normal cases.

Configuring the Interval of Sending PIM-DM State Refresh Message

After PIM-DM is enabled on a device, if an interface is directly connected to the multicast source, the device periodically sends state refresh messages to downstream devices to update the entire network state. You can modify the interval of sending PIM state refresh message on an interface according to the network situation.

Use the following command to configure the interval of sending PIM state messages on the interface in interface configuration mode.

Command	Function
Ruijie(config-if)#**ip pim state-refresh origination-interval** *seconds*	Configures the interval of sending PIM state refresh messages on the current interface, in the range of 1 to 100 seconds.
Ruijie(config-if)#**no ip pim state-refresh origination-interval**	Restores the setting to the default value.

By default, the interval of sending PIM state refresh messages on the interface is 60 seconds.

Note

Only the devices directly connected to multicast source can periodically send the PIM state refresh message to the downstream interfaces. Therefore, if the devices are not directly connected to the multicast source, the interval of sending PIM state refresh messages configured on the downstream interface is invalid.

Monitoring and Maintaining PIM-DM

PIM-DM provides the following commands to monitor and maintain PIM-DM.

Displaying PIM-DM State

Command	Function
show ip pim dense-mode interface [*interface-type interface-number*] [**detail**]	Shows the PIM-DM information on the interface.
show ip pim dense-mode neighbor [*interface-type interface-number*]	Shows the PIM-DM neighbor information.
show ip pim dense-mode nexthop	Shows the next hop information of PIM-DM.
show ip pim dense-mode mroute [*group-or-source-address [group-or-source-address*]] [**summary**]	Shows the PIM-DM routing table.
show ip pim dense-mode track	Shows the number of PIM packets sent and received since the statistic beginning time.

For details about the preceding commands, see *PIM-DM Command References*.

Here are some examples of the commands:

show ip pim dense-mode interface detail command:

```
Ruijie# show ip pim dense-mode interface detail
FastEthernet 0/1 (vif-id: 3):
Address 10.10.10.10
Hello period 30 seconds, Next Hello in 15 seconds
Over-ride interval 2500 milli-seconds
Propagation-delay 500 milli-seconds
Neighbors:
10.10.10.1
FastEthernet 0/2 (vif-id: 2):
Address 50.50.50.50
Hello period 30 seconds, Next Hello in 2 seconds
Over-ride interval 2500 milli-seconds
Propagation-delay 500 milli-seconds
Neighbors:
50.50.50.1
```

In the preceding example, the IP address of FastEthernet 0/1 is 10.10.10.10, the Hello message sending interval is 30 seconds, the next Hello message is to be sent in 15 seconds, and the neighbor address is 10.10.10.1. The information about FastEthernet 0/2 is similar to that of FastEthernet 0/1.

show ip pim dense-mode neighbor command:

```
Ruijie# show ip pim dense-mode neighbor
Neighbor-Address Interface           Uptime/Expires    Ver
10.10.10.1      FastEthernet 0/1      00:19:29/00:01:21  v2
50.50.50.1      FastEthernet 0/2      00:22:09/00:01:39  v2
```

In the preceding example, the device has two neighbors. Neighbor 10.10.10.1 connects to FastEthernet 0/1 and has been alive for 19 minutes and 29 seconds, with the TTL to expire in one minute and 21 seconds. Neighbor 50.50.50.1 is similar.

show ip pim dense-mode nexthop command:

```
Ruijie# show ip pim dense-mode nexthop
Destination  Nexthop   Nexthop      Nexthop            Metric Pref
              Num      Addr        Interface
1.1.1.111      1      50.50.50.1    FastEthernet 0/2    0      1
```

In the preceding example, the next hop neighbor address to multicast source 1.1.1.111 is 50.50.50.1 and the egress is FastEthernet 0/2.

show ip pim dense-mode mroute command:

```
Ruijie# show ip pim dense-mode mroute
PIM-DM Multicast Routing Table
(1.1.1.111, 229.1.1.1)
MRT lifetime expires in 205 seconds
RPF Neighbor: 50.50.50.1, Nexthop: 50.50.50.1, FastEthernet 0/2
Upstream IF: FastEthernet 0/2
Upstream State: Pruned, PLT:200
Assert State: NoInfo
Downstream IF List:
FastEthernet 0/1:
```

```
Downstream State: NoInfo
Assert State: Loser, AT:170
```

The preceding example shows two entries: 1.1.1.111 and 229.1.1.1. The MRG aging time is 205 seconds, RPF neighbor is 50.50.50.1, the next hop is 50.50.50.1, and the egress to the next hop is FastEthernet 0/2. The upstream interface of these entries is FastEthernet 0/2 in Pruned state, indicating that there is no downstream forwarding egress. The downstream interface is FastEthernet 0/1 in NoInfo state. The Assert state of the interface is Loser. FastEthernet 0/1 is not a forwarding egress.

show ip pim dense-mode track command:

```
Ruijie# show ip pim dense-mode track
PIM packet counters
Elapsed time since counters cleared: 00:04:03

                           received    sent
Valid PIMDM packets:            1           8
Hello:                          1           8
Join/Prune:                 0           0
Graft:                          0            0
Graft-Ack:                      0            0
Assert:                     0            0
State-Refresh:              0            0
PIM-SM-Register:            0            0
PIM-SM-Register-Stop:           0            0
PIM-SM-BSM:                     0            0
PIM-SM-C-RP-ADV:                0            0
Unknown Type:                   0

Errors:
Malformed packets:          0
Bad checksums:              0
Unknown PIM version:            0
Send errors:                    0
```

Deleting PIM-DM State Information

Use the following command to delete the PIM-DM state information:

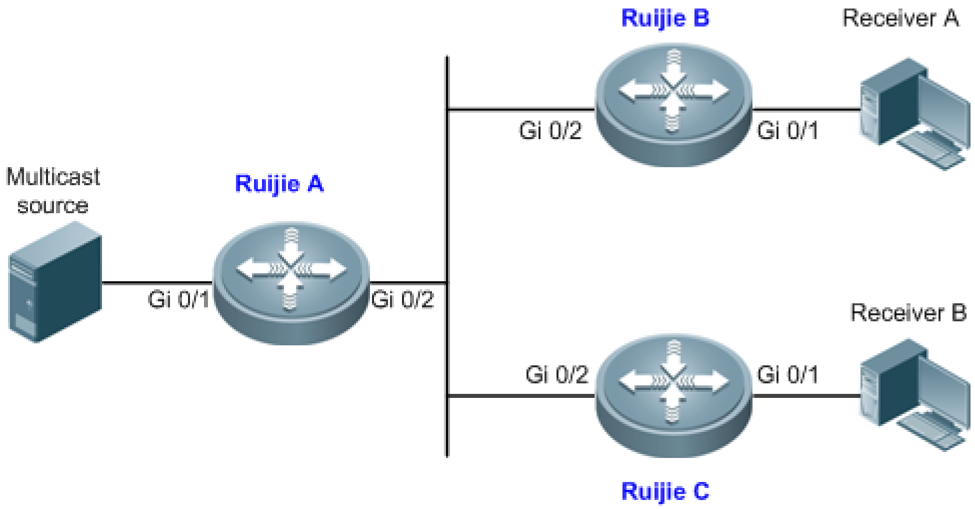
Command	Function
Ruijie# **clear ip pim dense-mode track**	Resets the start time of packet statistics and clears PIM packet counter.

PIM-DM Configuration Example

Configuration Requirements

Figure 12 shows the network topology. Ruijie A and the multicast source are in the same network, Ruijie B and receiver A are in the same network, and Ruijie C and receiver B are in the same network. Assume that the devices are properly connected to the hosts, and the IP addresses and unicast routes are configured.

Figure 12 Networking topology for PIM-DM configuration



Device Configuration

The following example shows how to configure PIM-DM on Ruijie A. The configurations on Ruijie B and Ruijie C are similar to those on Ruijie A.

- Step 1: Enable multicast routing.

```
Ruijie# configure terminal
Ruijie(config)# ip multicast-routing
```

- Step 2: Enable PIM-DM on the interface Gi 0/1.

```
Ruijie(config)# interface GigabitEthernet 0/1
Ruijie(config-if)# ip pim dense-mode
Ruijie(config-if)# exit
```

- Step 3: Enable PIM-DM on the interface Gi 0/2 and return to the privileged EXEC mode.

```
Ruijie(config)# interface GigabitEthernet 0/2
Ruijie(config-if)# ip pim dense-mode
Ruijie(config-if)# end
```

The configuration on Ruijie B and Ruijie C is similar to Ruijie A, that is, enable multicast routing and enable PIM-DM on each interface.

Note

Enabling PIM-DM will automatically enable IGMP on each interface.

This example applies to both layer-3 switches and routers. However, if you need to configure the switch port as a layer-3 port, you must run the **no switchport** command (not needed for routers).

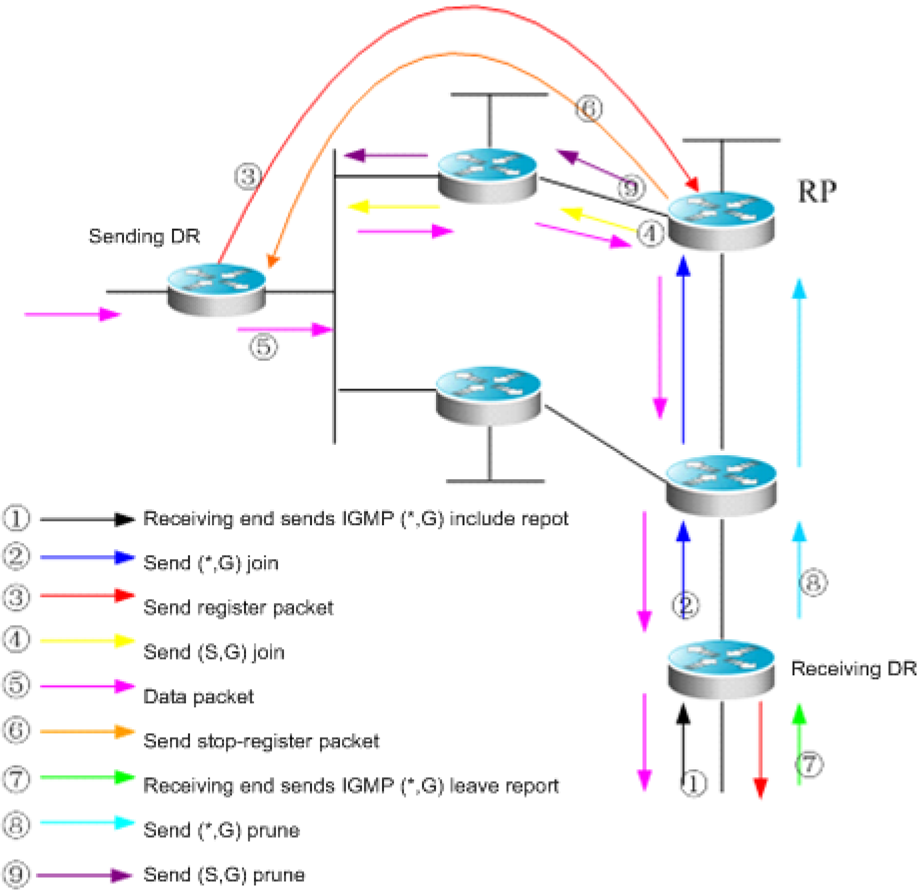
Configuring PIM-SM

PIM-SM Overview

The Protocol Independent Multicast (PIM) is designed by the Inter-Domain Multicast Routing (idmr) working group. As its name implied, PIM does not rely on any specific unicast routing protocol. It can use a unicast routing table established by any unicast routing protocol to perform the RPF check function, instead of maintaining separate multicast routing tables to implement multicast forwarding. As PIM is not required to receive or distribute route updates, compared to other multicast routing protocols, it costs much less. PIM is designed to support shortest path trees (SPTs) and rendezvous point trees (RPTs) simultanenously and enable flexible conversion between them, so that their advantages can be used to improve multicast efficiency. There are two PIM modes: dense mode and sparse mode.

The Protocol Independent Multicast – Sparse Mode (PIM-SM) is a multicast routing protocol of sparse mode. In a PIM-SM domain, the PIM-SM-enabled device periodically sends Hello messages to discover adjacent PIM-SM devices and selects a designated router (DR) in a multi-access network. The DR is responsible for sending Join/Prune messages towards the root of themulticast distribution tree from its directly connected group member, or itsdirectly connected multicast source.

Figure 13 Explicit Join/Prune mechanism of PIM-SM



PIM-SM forwards multicast data packets by establishing a multicast distribution tree. The multicast distribution tree is divided into two types: Shared Tree that takes the RP of the group G as the root and Shortest Path Tree that takes the multicast source as the root. PIM-SM establishes and maintains the multicast distribution tree by use of the explicit join/prune mechanism. As shown in Figure-1,

The DR at the receiving end receives an IGMP (*,G)include report packet from the receiving end.
If the DR at the receiving end is not the RP of this group G, it will send a (*.G)join packet towards the RP. The upstream router receiving this (*.G)join packet will send it towards the RP. In this way, the (*.G)join packet is sent hop by hop until the RP of the group G receives the (*.G)join packet. It is indicated that the DR has joined the shared tree.
When the source host sends multicast data to the group, the source data is encapsulated into a register message and unicasted by the DR at the data source to the RP. Then the RP will forward the decapsulated data packets to group members along the shared tree.
The RP will send a (S, G)join packet to the DR in the direction of the source to join the shortest path tree of this source. In this way, the source's packets are sent to the RP without encapsulation along its shortest path tree after the SPT from the RP to the DR at the source is established.
When the first multicast data reaches along the SPT, the RP will send the register - stop message to the DR at the source, notifying the DR of stopping register encapsulation. When the DR at the source received the register - stop message, it will not encapsulate register packets, but send them to the RP along its shortest path tree, which will forward them to group members along the shared tree.
When a receiving end needs no multicast data, it will send an IGMP leave message.
The DR at the receiving end multicasts the prune message to the group G's RP hop by hop to prune the shared tree. This prune message will finally arrive at the RP or a router with other (*G) receivers on the way to the RP. Therefore, the data packets will not be sent toward that receiving end.
If there is no downstream receiver on the RP, the RP will send the (S,G) prune packet toward the data source. As the (S, G) prune packets are sent to the DR at the source end one by one, the DR at the source end will prune the interface receiving the (S,G) prune packet. As a result, the data packets are filtered at the DR at the source end.

PIM-SM also offers a mechanism of selecting the root point (RP). One or more Candidate-BSRs are configured in a PIM-SM domain. PIM-SM selects a BSR by following a certain rule. There are also Candidate-RPs in a PIM-SM domain that unicast the packets including their IP addresses and available multicast groups to the BSR. The BSR will periodically generate a BSR message which includes a series ofcandidate RPs and corresponding multicast group addresses. The BSR messages are sent hop-by-hop within the entire domain. The device receives and saves these BSR messages. If the DR receives a report on the member relationship of a multicast group from its directly connected host but has no route entries of the multicast group, the DR will use a Hash algorithm to map the multicast group address to a candidate RP that can serve this group. Then, the DR multicasts the Join/Prune message to the RP hop-by-hop. If the DR receives multicast data packets from its directly connected host but has no route entries of the multicast group, the DR will usea Hash algorithm to map the multicast group address to a candidate RP that can serve this group. Then the DR encapsulates multicast data packets into a register message and unicasts it to the RP.

The main difference between PIM-SM and the flood/prune model-based PIM-DM is that PIM-SM is based on the explicit join model. In other words, the receiver sends the join message to the RP, while the router only forwards the packets of that multicast group on the outgoing interface that has joined a multicast group. PIM-SM uses the shared tree to forward multicast packets. Each group has a Rendezvous Point (RP). The multicast source sends data to the RP along the shortest path, and then the RP sends the data to the receivers along the shortest path. This is similar to CBT, but PIM-SM does not use the concept of core. One of the major advantages of PIM-SM is that it not only receives multicast messages through the shared tree but also provides a shared tree-to-SPT conversion mechanism. Such conversion reduces network delay and possible congestion on the RP, but it consumes enormous router resources. So it is suitable for the case where there are only a few multicast data sources and network groups.

PIM-SM uses the shared tree and SPT to distribute multicast frames. At this time, it is assumed that other devices don't want to receive these multicasts unless otherwise stated definitely. When a host joins a group, the equipment connected to the host must notify the root (or the RP) by using the PIM join message. This join message is transferred one after another through the routers to create a shared tree structure. Therefore, the RP records the transfer path and also the register message from the first hop router (DR) of the multicast source, and improves the shared tree upon these two messages. The branch/leaf messages are updated by periodically querying messages. With the shared tree, the multicast source first sends multicast packets to the RP, guaranteeing that all the receivers can receive them. The notation (*.G) represents a tree. The asterisk (*) represents all sources and G represents a specific multicast address. The prune message is also used in the shared tree. That is, the branch/leaf will send prune messages once it is not expecting to receive multicast frames.

PIMv2 BSR is a method of distributing group-to-RP messages to all devices without the need of setting an RP for them. BSR distributes mapping information by propagating BSR messages hop by hop. At first, BSR is selected among routers in the same process as selecting a root bridge based on priority level among layer 2 bridges. Each BSR checks the BSR messages and only forwards those having a priority higher than or equal to its own (higher IP address). The selected BSR sends its BSR message to the all-PIM-routers multicast group (224.0.0.13), where TTL is 1. After the adjacent PIMv2 router receives the message, it multicasts it while setting the TTL to 1. In this way, the BSR message is received by all devices hop by hop. Since the message contains the IP address of the BSR, the candidate BSR can know which router is the current BSR based on this message. The candidate RPs send candidate RP advertisements to announce in which address ranges they can become an RP. The BSR stores them in its local candidate RP cache. The BSP notifies all PIM routers of its local candidate RPs periodically. These messages reach various devices hop by hop in the same way.

☑ The VRF parameters only apply to RSR20, RSR30, RSR50 and RSR50E

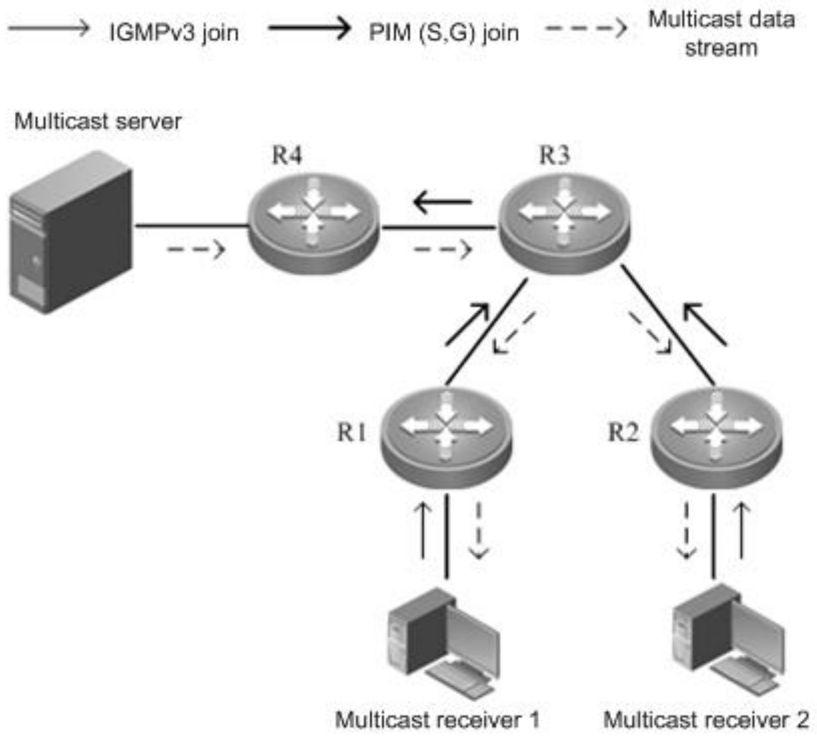
SSM Model

PIM-SM allows two multicast models,: Any-Source Multicast (ASM) and Source-Specific Multicast (SSM). In the ASM model, multicast receivers only specify a multicast group G to join but not a multicast source S. In the SSM model, multicast receivers can specify both a multicast source S and multicast group G.

The PIM-based SSM model provides implementation solutions for specified source multicast. It requires IGMPv3 to manage the membership between hosts and routers and PIM-SM to connect routers.

In the SSM model, multicast receivers have known the multicast source information (S, G) by some means such as accessing the server and accepting advertisements. Then, when a multicast receiver requests a multicast service, it can send IGMP(S, G) join directly to the last hop router. As shown in Figure 2, multicast receiver 1 sends the IGMP(S, G) join report to request the multicast service (S, G). The last hop router sends PIM (S, G) join to the multicast source hop by hop after receiving the IGMP (S, G) join from the multicast receiver. As shown in Figure 2, R1 sends the PIM(S, G) join to R3 after receiving the IGMP(S, G) join report from multicast receiver 1, and then R3 sends the PIM(S, G) join to R4. In this way, a shortest path tree from the multicast receiver to the multicast source is created.

Figure 14 SSM Model



Implementation of an SSM model requires that,

- The multicast receiver obtains the information about the multicast source (S, G) in advance through some channel; the multicast receiver initiates IGMP(S, G) join for the desired multicast services.
- IGMPv3 must be enabled on the interface of the last hop router connected to the multicast receiver. IGMPv1/IGMPv2 does not support the SSM.
- It is recommended that PIM-SSM be enabled on the routers on the way from the multicast receiver to the multicast source. AS PIM-SSM is compatible with PIM-SM, it is feasible to enable only PIM-SM.

Note With SSM enabled, the default group range of SSM is 232/8. The group range of SSM can be modified through commands or SSM can be disabled. For details, see the "Configuring SSM" section.

The SSM has the following features:

- In the SSM model, multicast receivers can obtain the information about the multicast source in advance through some channels, for example, advertisements or access to the specified server.
- The SSM model is a specific subset of PIM-SM. It only processes the PIM(S,G) join and PIM(S,G) prune messages and drops the RPT-related messages within the range of SSM, e.g. PIM(*,G) join/prune messages. It will respond immediately with the register - stop packet to the register packets within the range of SSM.
- In the SSM model, no PR or the election and distribution of RP messages is required. In the SSM, all the multicast distribution trees created are the shortest path trees (SPT).

Preparation before Configuring PIM-SM

Before configuring PIM-SM, enable a routing protocol such as OSPF to automatically discover routes.

PIM-SM Configuration Tasks

The PIM-SM configuration tasks cover the following items. However, only the first and second one are mandatory, and others are optional.

Enabling Multicast Routing

PIM-SM can forward multicast packets only after the multicast routing function is enabled.

Use the following command to enable the multicast routing function in global configuration mode.

Command	Function
ip multicast-routing [**vrf** *vrf-name*]	Enables multicast routing. If VRF is carried, multicast routing is enabled based on the VRF; if VRF is not carried, multicast routing is enabled globally.
no ip multicast-routing [**vrf** *vrf-name*]	Disables multicast routing.

Enabling PIM-SM

PIM-SM must be enabled on each interface. A device can exchange PIM-SM control messages with other devices, maintain and update multicast routing table, and forward multicast packets only after PIM-SM is enabled on its interface.

Use the following command to enable PIM-SM on the interface in interface configuration mode.

Command	Function
ip pim sparse-mode	Enables PIM-SM on the interface.
no ip pim sparse-mode	Disables PIM-SM on the interface.

Note

PIM-SM can be enabled on an interface only after multicast routing is enabled in global configuration mode.

Note

If the system prompts "Failed to enable PIM-SM on <*interface name*>, resource temporarily unavailable, please try again", run this command again.

Note

If the system prompts "PIM-SM Configure failed! VIF limit exceeded in NSM!!!", the number of configured interfaces exceeds the upper limit of the multicast interfaces. Remove some unnecessary PIM-DM, PIM-SM or DVMRP interfaces.

Note

It is not recommended that different IPv4 multicast protocols be configured on different interfaces of a switch or router.

Note

If the interface is of tunnel-type, only 4Over4 configuration tunnel, 4Over4 GRE tunnel, 4Over6 configuration tunnel and 4Over6 GRE tunnel support the IPv4 multicasting. Multicast can be also enabled on other tunnel interfaces that do not support the multicasting, but no error message will be displayed and no multicast packets will be received or sent.

Note

The multicast tunnel can be created on the Ethernet interface only. Nested tunnel and multicast data QoS/ACL are not supported.

Configuring the Interval of Sending Hello Messages

After PIM-SM is enabled on the interface, the device periodically sends Hello messages to the interfaces of neighbors. You can set the interval of sending Hello messages according to the actual network situation.

Use the following command to configure the interval of sending the Hello message in interface configuration mode.

Command	Function
ip pim query-interval *interval-seconds*	Sets the interval of sending Hello messages, in seconds. *interval-seconds*: in the range of 1 to 65535 seconds
no ip pim query-interval	Restores the setting to the default value.

By default, the interval of sending Hello messages on the interface is 30 seconds.

Note

When the interval of sending Hello messages is updated, Hello hold time will automatically be updated to 3.5 times of the interval of sending Hello messages. If the result is greater than 65535, Hello hold time is updated to 65535.

Configuring Propagation-Delay in Hello Message Option

Options can be added to Hello messages. The default value of propagation-delay in LAN Prune Delay Option is 500 milliseconds.

Use the following command to configure the propagation delay of Hello messages in interface configuration mode.

Command	Function
ip pim propagation-delay *interval-milliseconds*	Sets the propagation delay in the range of 1 to 32767 milliseconds.
no ip pim propagation-delay	Restores the setting to the default value.

Note

Modifying propagation delay or prune deny delay will affect J/P-override-interval. As specified in the protocol, J/P-override-interval is less than the hold time of Join-Prune message; otherwise streams may be interrupted temporarily. This can be ensured by network administrators.

Configuring Override-Interval in Hello Message Option

Options can be added to Hello messages. The default value of override-interval in LAN Prune Delay Option **i**s 2500 milliseconds.

Use the following command to configure the override interval in interface configuration mode.

Command	**Function**
ip pim override-interval *interval-milliseconds*	Sets the override interval in the range of 1 to 65535 milliseconds.
no ip pim override-interval	Restores the setting to the default value, namely, 2500 milliseconds.

Note

Modifying propagation delay or prune deny delay will affect J/P-override-interval. As specified in the protocol, J/P-override-interval must be less than the hold time of Join-Prune message; otherwise streams may be interrupted temporarily. This can be ensured by network administrators.

Configuring Neighbor-Tracking in Hello Message Option

The T bit of the LAN Prune Delay Option of the Hello message indicates whether to enable join restriction on the interface. When join restriction is enabled on the interface, the Join message to be sent from the interface to the upstream neighbor will be restricted, upon receipt of the Join message from its neighbor to the upstream neighbor. If this function is disabled, the Join message to be sent from the interface to the upstream neighbor will still be sent. Moreover, if join restriction is enabled on all downstream receivers, the upstream router can trace these receivers by received Join messages. By default, join restriction is enabled on the interface.

Use the following command to disable join restriction on the interface in interface mode.

Command	**Function**
ip pim neighbor-tracking	Disables join restriction on the interface.
no ip pim neighbor-tracking	Enables join restriction on the interface.

Configuring Triggered Hello Delay of Hello Messages

When a router starts or detects new neighbor, the device will send Hello messages after a random period of time to prevent congestion of Hello packets. This random interval can be calculated based on triggered hello delay, which is 5 seconds by default.

Use the following command to configure the triggered Hello delay in interface configuration mode.

Command	**Function**
ip pim triggered-hello-delay *interval-seconds*	Sets the triggered hello delay, in seconds. *interval-seconds*: in the range of 1 to 5 seconds
no ip pim triggered-hello-delay	Restores the setting to the default value.

Configuring PIM-SM Neighbor Filtering

You can filter neighbors on an interface to enhance network security. With this function enabled, when a neighbor is denied by an ACL, the PIM-SM will not establish the adjacency relationship with that neighbor or remove the currently established adjacency relationship with that neighbor.

Use the following command to configure PIM-SM neighbor filtering in interface configuration mode:

Command	Function
ip pim neighbor-filter *access-list*	Enables the PIM-SM neighbor filtering function on the interface.
no ip pim neighbor-filter *access-list*	Disables the PIM-SM neighbor filtering function on the interface.

By default, the PIM-SM neighbor filtering function is disabled on an interface.

Note

ip pim neighbor-filter command description:
When the associated ACL rule is permit, only the neighbor address in the ACL list can be used as the PIM neighbor of the current interface; when the associated ACL rule is deny, the neighbor address in the ACL list cannot be used as the PIM neighbor of the current interface.

Configuring the Priority of DR

This command is used to configure the priority of the designated router (DR). Higher weight means higher priority.

Use the following command to configure the DR priority in interface configuration mode.

Command	Function
ip pim dr-priority *priority-value*	Sets the DR priority. *priority-value*: in the range of 0 to 4294967294
no ip pim dr-priority	Restores the setting to the default value, namely 1.

Configuring Static RP

In a small network, you can configure static RP to use PIM-SM. All the devices in the PIM-SM domain have the same static RP configuration, ensuring no ambiguity of the PIM-SM multicast routes.

Use the following command to configure static RP in global configuration mode.

Command	Function
ip pim [**vrf** *vrf-name*] **rp-address** *rp-address* [*access-list*]	Configures the static RP address. *access-list*: supports the numerical ACL in the range of 1 to 99 and 1300 to 1999; also supports the named ACL. All multicast groups are permitted by default.
no ip pim [**vrf** *vrf-name*] **rp-address** *rp-address* [*access-list*]	Removes the static RP configuration.

Note

If the static RP and the dynamic RP take effect at the same time, the dynamic RP takes precedence. .

Note

The static RP address can be configured for multiple multicast groups (by ACL) or all multicast groups (not by ACL). However, a static RP address cannot be configured for several times.

Note

If more than one static RP are configured for a multicast group, the one with the highest IP address takes effect.

Note

Only the permitted addresses defined in the ACL are invalid multicast groups. By default, 0.0.0.0/0 refers to filtering all multicast groups (224/4).

Note

After configuration, the static RP source address is inserted into the tree of group-based static RP group. Each static multicast group maintains the link table structure of a static RP group. The link tables are ordered in descending sequence by IP addresses. When a RP is selected for a group, the first element, namely, the RP with the highest IP address is firstly selected.

Note

Deleting a static RP address deletes the address from all groups that has this address, and one address is selected from the existing tree structure as the RP address.

Configuring Candidate BSR

A globally unique BSR is elected from candidate BSRs configured on an interface in a PIM-SM domain. This BSR will collect and distribute RPs in the domain to ensure the uniqueness of RP mapping in the domain.

Use the following command to configure the candidate BSR in global configuration mode.

Command	Function
ip pim [**vrf** *vrf-name*] **bsr-candidate** *interface-type interface-number* [*hash-mask-length* [*priority-value*]]	Specifies an interface as a candidate BSR. The global BSR is elected through BSM message learning and competition. *hash-mask-length* ranges from 0 to 32, 10 by default. *priority-value* ranges from 0 to 255, 64 by default.
no ip pim [**vrf** *vrf-name*] **bsr-candidate** *interface-type interface-number*	Removes the configuration of a candidate BSR.

Configuring BSR Border

To restrict BSM flooding, you can set the BSR border on the interface so that BSM will be dropped immediately rather than being forwarded.

Use the following command to configure the BSR border in interface configuration mode.

Command	Function
ip pim bsr-border	Configures the BSR border on an interface.
no ip pim bsr-border	Removes the BSR border on an interface.

Ignoring RP Priorities in RP-SET

When an RP is selected for a multicast address, if several RPs can serve this multicast address, you can use this command to ignore the RP priority when comparing two RPs. If this command is not configured, the RP priority will be taken into account during comparison.

Command	Function
ip pim [**vrf** *vrf-name*] **ignore-rp-set-priority**	Ignores the RP priority in RP-Set.
no ip pim [**vrf** *vrf-name*] **ignore-rp-set-priority**	Considers the RP priority in RP-Set.

Configuring Candidate RP

Candidate RP advertisement is sent to the BSR at intervals and then propagated to all the PIM-SM devices in the domain, thus ensuring the uniqueness of RP mapping.

Use the following command to configure the candidate RP in global configuration mode.

Command	Function
ip pim rp-candidate *interface-type interface-number* [**priority** *priority-value*] [**interval** *interval-seconds*] [**group-list** *access-list*]	Configures the device as the candidate RP. When **priority** is default, *priority-value* ranges from 0 to 255, 192 by default. When **interval** is default, *interval-seconds* ranges from 1 to 16383, 60 seconds by default. When **group-list** is default, *access-list* permits all multicast groups by default, namely 224/4.
no ip pim rp-candidate *interface-type interface-number*	Removes the candidate RP configuration.

Note

You can use the ACL to specify an interface as the candidate RP of a specific group. It should be noted that the group calculation is based on the permit ACE, not the deny ACE. The source range of the ACE is matched as a specific group range.

Checking Reachability of Register Messages

You can use this command to check whether an RP is reachable. With this command configured, the DR checks whether RP is reachable before sending a register packet, that is whether there is a route to the RP by checking the unicast and static multicast routing tables. If there is no such a route, no register packet will be sent.

Use the following command to check the RP reachability in global configuration mode.

Command	Function
ip pim [**vrf** *vrf-name*] **register-rp-reachability**	Checks whether a register packet can reach the destination device. No check is conducted by default.
no ip pim [**vrf** *vrf-name*] **register-rp-reachability**	Disables this function.

Note

If there is a static multicast route to the RP and the next hop of the route is reachable in the unicast routing table, PIM-SM considers that a route to the RP exists even if the RP is not reachable in the unicast routing table.

Configuring Address-based Filtering for Register Packets

You can use this command to filter the register packets that have arrived at an RP by the source address and group address contained in the packets. Otherwise, every reached register packet is permitted. With this command configured, only the register packets with the source addresses and group addresses permitted by the ACL can be processed.

Use the following command to configure address-based filtering for register packets in global configuration mode.

Command	Function
ip pim [**vrf** *vrf-name*] **accept-register list** *access-list*	Enables filter of register packets by source addresses and group addresses. *access-list* ranges from 100 to 199 and 2000 to 2699; also supports named ACL.
no ip pim [**vrf** *vrf-name*] **accept-register**	Disables filter of register packets by source addresses and group addresses.

Configuring Rate Limit on Sending Register Packets

Use this command to configure the rate of sending register packets by DR. Use the **no** form of this command to cancel the rate limit. This command configures the rate of sending register packets for each (S, G) state, not the register packets in the whole system.

Use the following command to configure the rate limit on sending RPs in global configuration mode.

Command	Function
ip pim [**vrf** *vrf-name*] **register-rate-limit** *rate*	Sets the maximum number of register packets sent per second. *rate*: in the range of 1-65535
no ip pim [**vrf** *vrf-name*] **register-rate-limit**	Removes the rate limit.

Configuring the Whole-Packet Method for Calculating the Register Packet Checksum

Use this command to calculate the whole PIM packet including the multicast data packet encapsulated when calculating the register packet checksum. Otherwise, the checksum of register packets is calculated using the default method specified by the protocol.

Use the following command to configure the whole-packet method for calculating the register packet checksum in global configuration mode.

Command	Function
ip pim [**vrf** *vrf-name*] **register-checksum-wholepkt** [**group-list** *access-list*]	Configures the whole-packet method for calculating the register packet checksum. group-list *access-list*: Apply this configuration to all multicast addresses by default.
no ip pim [**vrf** *vrf-name*] **register-checksum-wholepkt** [**group-list** *access-list*]	Removes the whole-packet method for calculating the register packet checksum. group-list *access-list*: By default, this configuration is removed for all the multicast addresses.

Note

Some devices from other vendors make checksum calculation of register packets based on the overall packets. This function is introduced in Ruijie's devices to be compatible with those devices.
If a device from those vendors serves as the RP and Ruijie's device serves as the source DR, you can use this command on the source DR; if the device from other vendors serves as the source DR and Ruijie's device serves as the RP, you can use this command on the RP.

Configuring the RP to Forward Multicast Packets to Downstream Interfaces after Decapsulating Register Packets

Use this command to decapsulate a register packet and forward its multicast packets. Without this command, the multicast packets in the register packet are not de-capsulated or forwarded.

Use the following command in global configuration mode.

Command	Function
ip pim [**vrf** *vrf-name*] **register-decapsulate-forward**	Decapsulates the register packet and forwards its multicast packets
no ip pim [**vrf** *vrf-name*] **register-decapsulate-forward**	Removes the configuration.

Note

Since the register packet is decapsulated and its multicast packets are forwarded through software, in case of decapsulation and forwarding of many register packets, this function incurs additional workload to CPU. So it is not recommended.

Limiting the Range of Legal BSRs

Use this command to limit the range of legal BSRs. Without this function, PIM-SM-enabled routers will receive all external BSM messages.

Use the following command in global configuration mode.

Command	Function
ip pim [**vrf** *vrf-name*] **accept-bsr list** *accept-bsr list*	Filters the BSM packet of BSR. The range is from 1 to 99, 1300 to 1999, or can be characters.
no ip pim [**vrf** *vrf-name*] **accept-bsr**	Removes the filtering of the BSM packet of BSR.

Note

This command filters the BSR address field of the BSM message. If this address is denied by ACL, the BSM message is filtered.

Configuring the Electing BSR to Limit the Legal CRP Address Range and the Multicast Group Range It Serves

Use this command to configure the electing BSR to limit the legal CRP address range and the multicast group range it serves. Without this function, the electing BSR will receive all external advertisement messages of candidate RPs.

In this command, the *source* parameter of ACL rule specifies the C-RP address and the *destination* parameter specifies the multicast group range the C-RP serves. If both addresses are denied by ACL, the group of the C-RP will be filtered.

Use the following command in global configuration mode.

Command	Function
ip pim [**vrf** *vrf-name*] **accept-crp list** *accpet-crp list*	Enables the electing BSR to filter the candidate RP advertisement. The range is from 100 to 199, 2000 to 2699 or can be characters.
no ip pim [**vrf** *vrf-name*] **accept-crp**	Disables the electing BSR to filter the candidate RP advertisement.

Configuring the Electing BSR to Receive the C-RP-ADV Message Whose Prefix-count Is 0

Use this command to configure the electing BSR to receive the C-RP-ADV message whose prefix-count is 0. Without this command, the electing BSR will not process the C-RP-ADV packet whose prefix-count is 0.

With this function, the electing BSR considers that the C-RP supports all groups after receiving the C-RP-ADV message whose prefix-count is 0.

Use the following command in global configuration mode.

Command	Function

Command	Function
ip pim [**vrf** *vrf-name*] **accept-crp-with-null-group**	Configures the electing BSR to receive the C-RP-ADV message whose prefix-count is 0.
no ip pim [**vrf** *vrf-name*] **accept-crp-with-null-group**	Removes the configuration.

Configuring the Source IP Address of Register Packets

This command sets the source IP address of register packets sent from DR. With this command not configured or the **no** form of this command, the DR interface address connected to the multicast source is used as the source address of the register packet. If the address parameter of this command is used, the configured address must be reachable for unicast routes. If the interface parameter of this command is used, it is generally a loopback interface, but can also be other types. This interface address must have been advertised by the unicast route.

Use the following command in global configuration mode.

Command	Function
ip pim register-source { *local_address* | *Interface-type interface-number* }	Configures the source IP address used in RPs.
no ip pim register-source	Sets the RPF interface address as the source IP address of register packets.

Configuring Register Suppression Time

This command configures the registersuppression time. It will modify the register suppression time defined on the DR. If the **ip pim rp-register-kat command** is not configured, defining the register suppression time in the RP will change RP keepalive period.

Use the following command to configure the register suppression time in global configuration mode.

Command	Function
ip pim [**vrf** *vrf-name*] **register-suppression** *seconds*	Configures the register suppression time. *seconds* ranges from 1 to 65535 seconds.
no ip pim [**vrf** *vrf-name*] **register-suppression**	Sets the suppression time to 60 seconds.

Configuring the Probe Interval of Null Register Packet

The source DR can send Null-Register packet to the RP in a period of time before the registersuppression time expires. This period is called probe interval, 5 seconds by default.

Use the following command to configure the probe time of null register packets in global configuration mode.

Command	Function
ip pim [**vrf** *vrf-name*] **probe-interval** *interval-seconds*	Configures the probe time of null register packets. *interval-seconds*: in the range of 1 to 65535 seconds

Command	Function
no ip pim [**vrf** *vrf-name*] **probe-interval**	Restores the probe time to 5s.

Note

The probe time should be less than half of register suppression time. Moreover, the register suppression time times three and plus the probe time should not be greater than 65535 seconds; otherwise, the system displays a warning message.

Configuring the RP KAT Timer

Use this command to configure the keepalive time of the (S, G) state created by register packets on the RP in global configuration mode.

Command	Function
ip pim [**vrf** *vrf-name*] **rp-register-kat** *seconds*	Configures KAT timer. *seconds*: in the range of 1 to 65535 seconds
no ip pim [**vrf** *vrf-name*] **rp-register-kat**	Uses the default KAT value, namely, register suppression time times three and plus register probe time.

Note

The value of the timer should be greater than register suppression time of source DR times three and plus register probe time. Otherwise, the RP may timeout the (S, G) state before the source DR sends the register packet again, causing temporary interruption of the multicast stream.

Configuring the Interval of Sending the Join/Prune Message

By default, the Join/Prune message is sent at the interval of 60 seconds by default. Use this command to modify this interval. With this command not configured, the default sending interval of join/prune packets is 60 seconds.

Use the following command in global configuration mode.

Command	Function
ip pim [**vrf** *vrf-name*] **jp-timer** *seconds*	Sets the interval of sending the Join/Prune message. *interval-seconds*: in the range of 1 to 65535 seconds.
no ip pim [**vrf** *vrf-name*] **jp-timer** [*interval-seconds*]	Restores the setting to the default value, namely 60s.

Note

When the sending interval of the join/prune packets is configured, if the interval times3.5 is more than 65535, a warning message is displayed and the interval is changed to 65535/3.5 seconds.

Allowing the Last Hop Device to Switch from the Shared Tree to the Shortest Path Tree

With this command configured, when the first (S, G) packet is received, a PIM join message is triggered and a source tree is created. If the keyword **group-list** is defined, all the groups specified will switch to the source tree. Use the **no** form of this command to enable the device to switch back to the shared tree and send a prune message.

Use the following command in global configuration mode.

Command	Function
ip pim [**vrf** *vrf-name*] **spt-threshold** [**group-list** *access-list*]	If **group-list** is defined, allows the last hop device of a specific group to switch from the shared tree to the shortest path tree. If **group-list** is not defined, allows all multicast groups.
no ip pim [**vrf** *vrf-name*] **spt-threshold** [**group-list** *access-list*]	Disables this function.

Configuring MIB in Dense Mode

Use this command to configure MIB in dense mode; With this command not configured, MIB is in sparse mode.

Use the following command in global configuration mode.

Command	Function
ip pim mib dense-mode	Uses MIB in dense mode.
no ip pim mib dense-mode	Uses MIB in sparse mode,

Enabling SSM

In the SSM mode, multicast packets can be directly received from the multicast source instead of through the RP tree.

Use the following command in global configuration mode.

Command	Function
ip pim [**vrf** *vrf-name*] **ssm** { **default** | **range** *access-list* }	Enables SSM.
no ip pim [**vrf** *vrf-name*] **ssm**	Disables SSM.

Monitoring and Maintaining PIM-SM

PIM-SM provides the following commands to monitor and maintain PIM-SM.

Displaying PIM-SM Information

Use the following commands to show information about PIM-SM on the local device.

Command	Function
show debugging	Shows the status of the debugging switch.
show ip pim sparse-mode [**vrf** *vrf-name*] **bsr-router**	Shows BSR details.
show ip pim sparse-mode [**vrf** *vrf-name*] **interface** [*interface-type interface-number*] [**detail**]	Shows the PIM-SM information of the interface.
show ip pim sparse-mode [**vrf** *vrf-name*] **local-members** [*interface-type interface-number*]	Shows local IGMP information about a PIM-SM interface.
show ip pim sparse-mode [**vrf** *vrf-name*] **mroute** [*group-or-source-address* [*group-or-source-address*]] [**proxy**]	Shows information about a PIM-SM multicast routing table. With the parameter **proxy**, this command shows the RPF Vector information about PIM-SM entries.
show ip pim sparse-mode [**vrf** *vrf-name*] **neighbor** [**detail**]	Shows information about the PIM-SM neighbors.
show ip pim sparse-mode [**vrf** *vrf-name*] **nexthop**	Shows the next hop of PIM-SM from NSM.
show ip pim sparse-mode [**vrf** *vrf-name*] rp-**hash** *group-address*	Shows a specified group address. *group-address* corresponds to RP information.
show ip pim sparse-mode [**vrf** *vrf-name*] **rp mapping**	Shows information about all current RPs and the groups they serve.
show ip pim sparse-mode [**vrf** *vrf-name*] **track**	Shows the number of PIM packets sent and received from the start time of statistics till now

☑ The parameter **proxy** in the command **show ip pim sparse-mode mroute** is only supported by RSR20, RSR30, RSR50 and RSR50E.

Deleting Internal Information About PIM-SM

Use the following commands to delete internal information about PIM-SM on the local device.

Command	Function
clear ip mroute [**vrf** *vrf-name*] { * | *group_address* [*source_address*] }	Deletes a multicast routing table entry.
clear ip mroute [**vrf** *vrf-name*] **statistics** { * | *group_address* [*source_address*] }	Deletes the statistics of a multicast routing table entry.
clear ip pim sparse-mode [**vrf** *vrf-name*] **bsr rp-set** *	Deletes RP-SET.
clear ip pim sparse-mode [**vrf** *vrf-name*] **track**	Resets the start time of statistics and clears the PIM packet counter

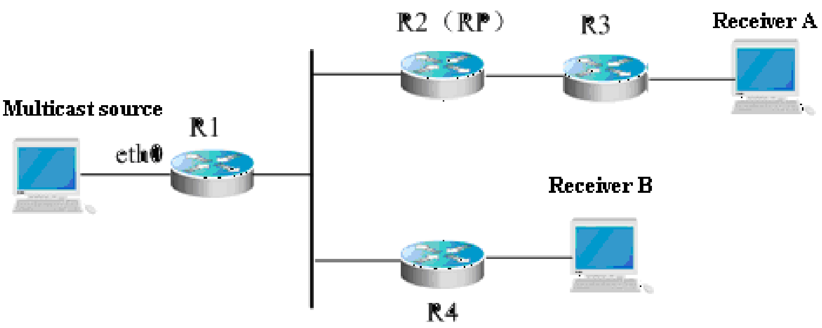
For details about the preceding commands, see *PIM-SM Commands*.

PIM-SM Configuration Example

Configuration Requirements

Figure 3 shows the network topology. R1 and the multicast source are in the same network. R2 will be set as an RP. R3 and receiver A are in the same network, and R4 and receiver B are in the same network. Assume that the devices connect to the host properly, IP addresses are configured on each interface, and IP unicast is enabled on each device.

Figure 15 Network topology for PIM-SM configuration



Device Configuration

Step 1: Enable multicast routing.

R1 is used as an example to show how to enable IP multicast routing. The configurations on R2, R3 and R4 are similar to R1.

```
Ruijie# configure terminal
Ruijie(config)# ip multicast-routing
```

Step 2: Enable PIM-SM on the interface.

The following shows how to enable PIM-SM on Gi 0/1 of R1. The configurations on interfaces of R1, R2, R3 and R4 are similar.

```
Ruijie(config)# interface GigabitEthernet 0/1
Ruijie(config-if)# ip pim sparse-mode
Ruijie(config-if)# end
```

Step 3: Configure the candidate BSR and the candidate RP.

Configure loopback 1 of R2 as C-BSR and C-RP.

```
Ruijie(config)# interface loopback 1
Ruijie(config-if)# ip address 100.1.1.1 255.255.255.0
Ruijie(config-if)# ip pim sparse-mode
Ruijie(config-if)# exit
Ruijie(config)# ip pim bsr-candidate loopback 1
Ruijie(config)# ip pim rp-candidate loopback 1
Ruijie(config-if)# end
```

After the receivers join the group and the multicast source sends multicast streams, you can use the **show** command provided by PIM-SM to monitor the running status.

Note Enabling PIM-DM will automatically enable IGMP on each interface.

Note This example applies to both layer-3 switches and routers. However, if you need to configure the switch port as a layer-3 port, you must run the **no switchport** command (not needed for routers).

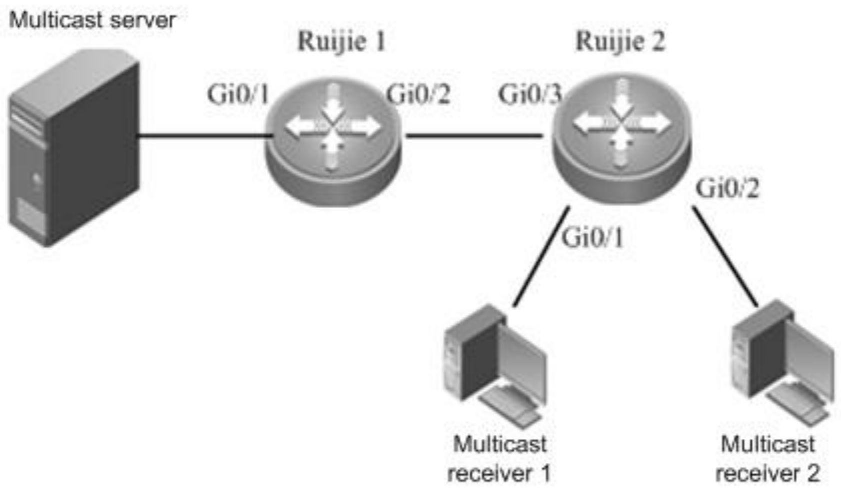
Enabling SSM based on PIM-SM

Networking Requirements

- The network must be interconnected on layer 3, for example, based on OSPF.
- The multicast receivers can obtain the information about the multicast source through some channels, depending on application software of hosts and deployment of network administrators.
- The PIM-SM protocol is applied within the network.

Networking Topology

Figure 16 Network topology for SSM configuration



Device name	Interface	IP address
Ruijie 1	Gi 0/1	2.2.2.1/24
	Gi 0/2	3.3.3.1/24
Ruijie 2	Gi 0/3	3.3.3.2/24
	Gi 0/1	4.4.4.1/24
	Gi 0/2	5.5.5.1/24

Configuration Steps

- Perform basic configuration on interfaces of Ruijie 1 and Ruijie 2.

Perform configurations on interfaces of Ruijie 1 and Ruijie 2 based on the IP addresses specified in the networking topology.

- Enable interworking on layer 3.

Enable OSPF on all interfaces of Ruijie 1 and Ruijie 2, implementing interworking on layer 3.

- Configure multicast on Ruijie 1 and Ruijie 2.

Firstly, enable the multicast routing on Ruijie 1 and Ruijie 2; secondly, enable PIM-SM on the interfaces of Ruijie 1 and Ruijie 2; at last, enable IGMPv3 on the interfaces where Ruijie 2 connects to the multicast receivers. Ruijie 2 is used as an example below.

# Enable the multicast routing on Ruijie 2.

```
Ruijie(conf)#ip multicast-routing
```

# Enable PIM-SM on all the interfaces of Ruijie 2. Interface Gi 0/1 is used as an example below:

```
Ruijie(conf-GigabitEthernet0/1)#ip pim sparse-mode
```

# Enable IGMPv3 on the interfaces where Ruijie 2 connects to the multicast receivers. Interface Gi 0/1 is used as an example below:

```
Ruijie(conf-GigabitEthernet0/1)#ip igmp version 3
```

The configuration procedure for Ruijie 1 is similar to that for Ruijie 2.

- Enable the SSM function on Ruijie 1 and Ruijie 2.

# Enable SSM on Ruijie 2. The default group range of SSM is used, namely 232/8.

```
Ruijie(conf)#ip pim ssm default
```

# Enable SSM on Ruijie 1. The default group range of SSM is used, namely 232/8.

```
Ruijie(conf)#ip pim ssm default
```

Verification

- Multicast receivers 1 and 2 request the multicast service (2.2.2.2, 232.0.0.1) by sending IGMP (2.2.2.2, 232.0.0.1) join.

Protocol entry (2.2.2.2, 232.0.0.1) will be created on Ruijie 2. Use the **show ip pim sparse-mode mroute** command to view it.

```
Ruijie#show ip pim sparse-mode mroute
IP Multicast Routing Table
(*,*,RP) Entries: 0
(*,G) Entries: 0
(S,G) Entries: 1
(S,G,rpt) Entries: 1
FCR Entries: 0
REG Entries: 0

(2.2.2.2, 232.0.0.1)
```

```
RPF nbr: 3.3.3.1
RPF idx: GigabitEthernet0/3
SPT bit: 0
Upstream State: JOINED
kat expires in 175 seconds
  00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29
30 31
Local
0.i  i  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
1  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
Joined
0  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
1  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
Asserted
0  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
1  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
Outgoing
0.o o  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
1  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
```

As shown in the preceding information, the protocol entry (2.2.2.2, 232.0.0.1) has been created on Ruijie 2 and there are two multicast receivers.

- The multicast server sends data stream (2.2.2.2, 232.0.0.1).

Multicast forwarding table (2.2.2.2, 232.0.0.1) will be created on Ruijie 2. Use the **show ip mroute** command to view it.

```
Ruijie#show ip mroute

IP Multicast Routing Table
Flags: I - Immediate Stat, T - Timed Stat, F - Forwarder installed, R - RPT, S - SPT
Timers: Uptime/Stat Expiry
Interface State: Interface (TTL)

(2.2.2.2, 232.0.0.1), uptime 00:19:31, stat expires 00:02:53
Owner PIMSM, Flags: TFSs
  Incoming interface: GigabitEthernet 0/3
  Outgoing interface list:
     GigabitEthernet 0/1(1)
     GigabitEthernet 0/2(1)
```

As shown in the preceding information, the multicast forwarding table (2.2.2.2, 232.0.0.1) has been created on Ruijie 2. Multicast egresses Gi 0/1 and Gi 0/2 are connected to multicast receivers 1 and 2 respectively. The forwarding table has been marked with the flag “s", which indicates that the table is in the SSM model.

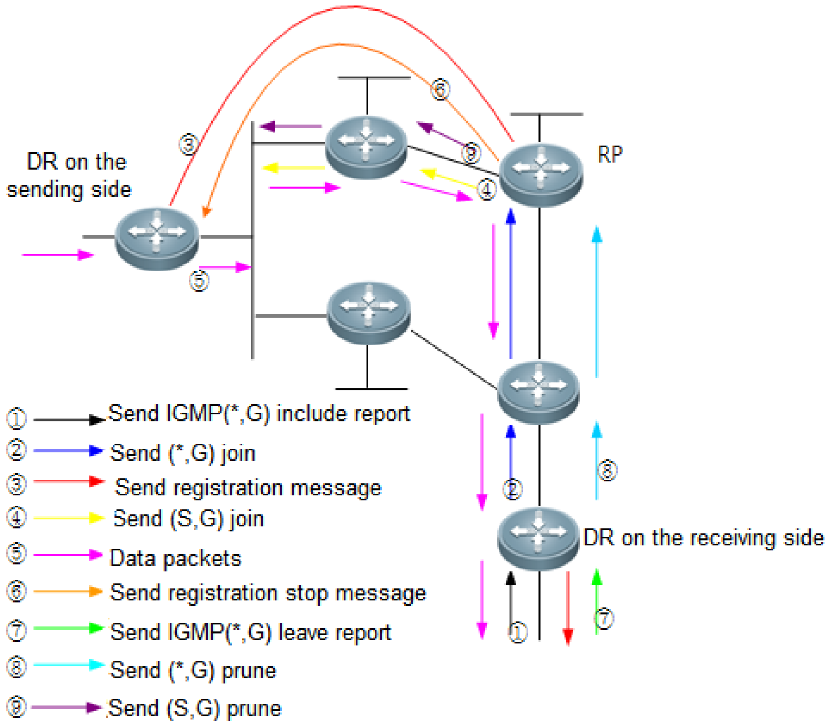
Configuring PIM-SMv6

PIM-SM Overview

PIM (Protocol Independent Multicast) is designed by IDMR (Inter-domain Multicast Routing) working group. As its name implied, PIM does not depend on a specific unicast routing protocol. It utilizes the unicast routing table established by various unicast routing protocols to enable the RPF check function instead of maintaining a separate multicast routing table for forwarding multicast packets. Compared with other multicast protocols, PIM overhead falls down at large extent for PIM does not need to receive and send multicast route update. The concept behind PIM design is that support and flexible transformation between SPT and the shared tree is enabled for higher multicast efficiency. There are two kinds of PIM modes-dense mode and sparse mode.

PIM-SM (Protocol Independent Multicast Sparse Mode) is a multicast routing protocol in sparse mode. In the PIM-SM domain, the device running PIM-SM sends the Hello message at a specific interval to discover adjacent devices running PIM-SM and be in charge of DR election. Here DR sends the “join/prune” message to its direct group members in the direction of the root node of the multicast distribution tree or sends the data from the direct multicast source to the multicast distribution tree.

Figure 17 PIM-SM explicit join/prune mechanism



PIM-SM forwards multicast data packets by setting up a multicast distribution tree. There are two kinds of multicast distribution tree: the shared tree using G’s RP as root and the shortest path tree using the multicast source as root. With the explicit join/prune mechanism, PIM-SM sets up and maintains the multicast distribution tree.

As illustrated in the above figure:

1. The DR on the receiving side receives the MLD (*,G) report message from a host on the same side.

2. If the DR on the receiving side is not the RP of G, it will send a (*.G) Join message to the RP. Upon receiving this message, the upstream router will send the (*.G) Join message to the RP without the forwarding entry of the corresponding group. In this way, the (*,G) Join message is transmitted hop to hop until G's RP receives the (*,G) Join message, indicating that the DR joins the shared tree.
3. When the source host sends multicast data to a group, the data is encapsulated in the registration message and sent by the DR on the source side to the RP in unicast form. Then, the RP decapsulates the registration message, extracts the data and forwards the data to every member of the group along the shared tree.
4. The RP sends the (S,G) Join message to the DR on the source side to join its shortest path tree.
5. After the shortest path tree from the RP to the DR on the source side is established, data packets are sent to the RP along this STP without encapsulation.
6. When the first multicast data arrives along the SPT, the RP sends the registration stop message to the DR on the source side, notifying the DR of stopping registration and encapsulation. Upon receiving the registration stop message, the DR no longer registers and encapsulates data packets. Instead, it sends data packets to the RP along the shortest path tree, which then forwards data packets to every group member along the shared tree.
7. When a receiving end does not need multicast packets, it sends the MLD leave message.
8. The DR on the receiving side sends the prune message in multicast form hop by hop to G's RP, the prune message arrives at the RP or the router along the way to the RP which has other (*,G) receiver so that data packets are not sent to this receiving side any more.
9. If the RP has no downstream receiver at present, it sends (S,G) prune message to the DR of the data source hop by hop. Consequently, the DR on the source side prunes the interface receiving this (S,G) prune message. In this way, data packets are filtered on the DR of the data source.

RP election is also involved in PIM-SM. when there is one or more candidate BSRs configured in the PIM-SM domain, some rule is applied to elect BSR. Candidate RPs are also configured in the PIM-SM domain, which send the packets containing their addresses and serviceable multicast groups to BSRs in unicast form. BSRs generate bootstrap messages with a series of candidate RPs and the addresses of corresponding multicast groups periodically. These bootstrap messages are transmitted in the overall domain hop by hop. Devices will receive and save these bootstrap messages. Upon receiving the member relation report of a multicast group from the directly connected device, the DR will use a hash algorithm and map the multicast group address to a candidate RP who can serve this group if it has not the routing entry of this group. Then the DR will send the join/prune message in multicast form hop by hop along the way to the RP. On the other hand, upon receiving multicast packets from the directly connected device, the DR will use a hash algorithm and map the multicast group address to a candidate RP who can serve this group, and then encapsulate these multicast packets in the registration message and send it to the PR in unicast form.

The essential difference between PIM-SM and PIM-DM is that PIM-SM is based on explicit join mode and PIM-DM is based on flood/prune mode. For PIM-SM, the receiver sends a join message to the PR, but the device forwards the packets of a multicast group only on the interface joining this multicast group. PIM-SM forwards multicast packets through the shared tree. Each multicast group has a rendezvous point. The multicast source sends packets to the RP along the shortest path, and then the RP sends the packets to every receiver along the shortest path. This process is similar to CBT. However, the core concept is not used in PIM-SMv6. One of the main advantages of PIM-SMv6 is that it not only receives multicast packets through the shared tree but also offers the shared tree-to-SPT transformation mechanism. This transformation consumes a lot number of resources, even though it reduces network delay and possible block on the RP. It is suitable for the environment where there are many pairs of multicast sources yet fewer networks.

PIM-SM distributes multicast packets through the shared tree and SPT. Assume that other devices do not need to receive these multicast packets, unless otherwise specified. When a host joins a multicast group, the devices connecting to the host notify the root (or RP) through the PIM join message. This message is transferred among these devices in order to set up the structure of a shared tree. So, the RP records this transmission path and the registration message from the first hop device (DR) of the sending multicast source, and perfects the shared tree based on these two messages. Update of leaf messages is enabled on periodic query message. For the shared tree, the multicast source sends multicast packets to the RP so that all receivers can receive these multicast packets. *.G indicates a tree, in which * indicates all sources and G indicates the specific multicast address. The prune message is also used in the shared tree when leafs do not need to receive multicast packets.

PIMv2 BSR distributes the group-to-RP message to all devices without the necessity for configuring RP for every device. The BSR distributes the mapping message through the hop-by-hop flooding BSR message. First of all, the BSR is elected among devices. This election procedure is similar to electing the root bridge in STP by priority. Every BSR device checks the BSR message, and only forwards the BSR messages with higher or equivalent priority (or higher IP address). The elected BSR sends the BSR message to the all-PIM-routers multicast group (ff02::d) with TTL 1. Upon receiving the BSR message, the adjacent PIMv2 device sends it out in multicast form and then reset TTL to 1. In this way, the BSR message is sent to all devices hop by hop. Since the BSR message includes the IP addresses of BSR devices, the candidate BSR can determine which device is the current BSR device. The candidate RP sends the candidate RP advertisement and alleges in which address ranges it can become RP. The BSR stores the advertisement message in its local candidate RP cache, and notifies all PIM devices of local candidate RPs periodically. Also in this way, the message is sent to all devices hop by hop.

Embedded RP is a new feature introduced in IPv6 multicast. It provides a new type of mapping from IPv6 multicast group to RP to replace the RP configured statically or the one calculated by BSR mechanism dynamically.

Some special IPv6 multicast addresses are called that of the embedded RP, the range of which is FF70::/12. You can obtain IPv6 unicast addresses by calculating the special IPv6 multicast group addresses, which means a RP address is embedded in each IPv6 multicast group address.

If the multicast group address extracted from MLD packets, PIM packets and multicast data packets by routers is the IPv6 multicast address of embedded RP, the router can calculate the RP address from this multicast group address directly. The embedded-RP multicast groups have the following advantages:

1) Since an IPv6 multicast group address is bound with a RP address. You can calculate the RP address directly rather than learning and updating the group RP mapping.

2) It can be deployed not only for intra-domain multicast, but also for inter-domain multicast. For IPv6 multicast group addresses of the same embedded RP, the same RP address is calculation no matter where the router is located.

SSM Model for PIM

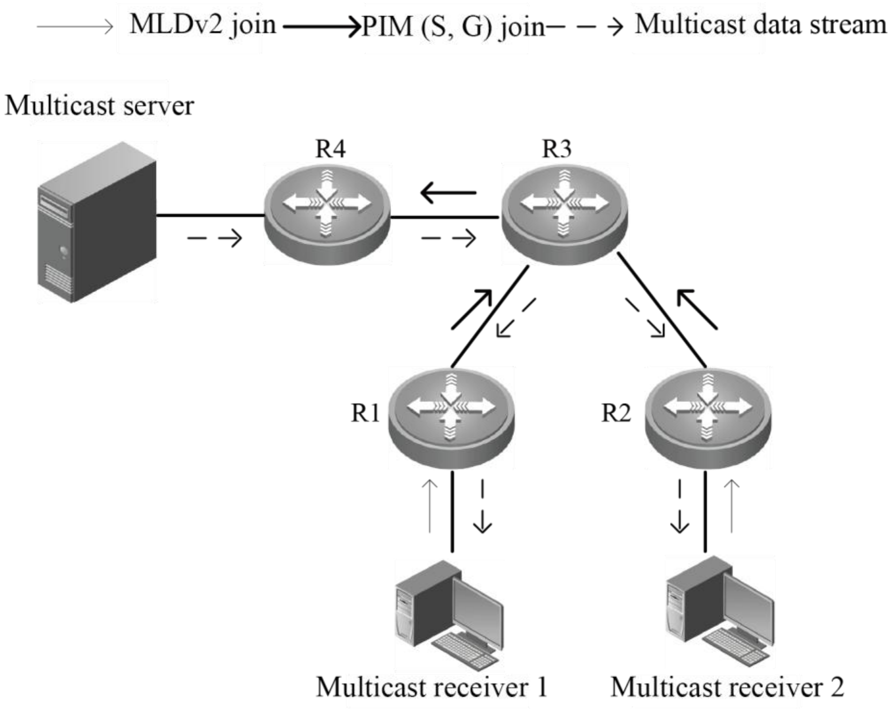
The traditional PIM-SM mode and PIM-DM mode are Any-Source Multicast (ASM) models, in which multicast data receivers specify only a specific multicast group G.

Source-Specific Multicast (SSM) and ASM are two independent models. The SSM model can be implemented by the PIM-SM technology, in which multicast data receivers specify both the multicast source S and multicast group G.

The PIM-based SSM model provides a solution for specifying the source multicast. It requires MLDv2 to manage the membership between hosts and routers and uses PIM-SM to connect hosts and routers.

In the SSM model, multicast receivers learn information about the multicast source (S,G) through a channel, such as accessing servers or accepting advertisements. Therefore, when the multicast receiver requests a multicast service, it can send the MLD (S, G) Join message directly to the last-hop router, as shown in Figure 2, and the multicast receiver 1 sends the MLD (S, G) Join message to request the multicast service (S, G). After receiving the MLD (S, G) Join message from the multicast receiver, the last-hop router sends PIM (S ,G) to the multicast source hop by hop, as shown in Figure 2. After receiving the MLD (S, G) Join message from multicast receiver 1, R1 sends PIM (S ,G) Join message to R3, and then R3 sends PIM (S ,G) Join message to R4. As a result, a shortest path tree from multicast receivers to the multicast source is established, as shown in Figure 2.

Figure 18 SSM working model



Requirements for SSM models:

1) The multicast receiver learns information about the multicast source (S, G) through a channel and initiates an MLD (S, G) Join message to the multicast service.

2) The last-hop router connects to the interface of multicast receivers and must be enabled with MLDv2 but not MLDv1. MLDv1 cannot processing the MLD Join message of the specified source.

3) It is recommended to enable PIM-SSM on path routers from multicast receivers to the multicast source. Since PIM-SSM is compatible with PIM-SM, you can enable only PIM-SM without enabling PIM-SSM.

By default, the multicast group range reserved for SSM is FF3x::/32. However, you can expand or shrink the multicast group range of SSM, or even disable the SSM function at the CLI. For more information, see the commands described in Configuring the Specific Source Multicast.

Compared to PIM-SM, PIM-SM-based SSM has the following features:

1) In the SSM model, multicast receivers can obtain information about the multicast source in advance through a channel, such as advertisements or accessing specified servers.

2) As a specific subset of PIM-SM, the SSM model processes only PIM (S, G) Join message and PIM (S, G) prune messages and discards RPT-associated messages that fall in the range of SSM, such as PIM (*, G) Join message and prune messages. For register messages fall into the range of SSM, the model will respond to the register-stop message immediately.

3) In the SSM model, RP information selection and distribution are unnecessary. The established multicast distribution tree in SSM are shortest path tree (SPT).

PIM-SMv6 Configuration Preparation

Enable a unicast routing protocol, for instance OSPFv3, to automatically discover routes.

PIM-SMv6 Configuration Task List

PIM-SMv6 configuration includes the following tasks, but only the first and the second tasks are mandatory, and you can determine whether to configure the other tasks based on actual networks.

Enabling multicase routing

Enabling PIM-SMv6

Configuring Hello message transmission interval

Configuring the propagation delay of the Hello message

Configuring the override interval of the Hello message

Configuring the neignbor tracking of the Hello message

Configuring the triggered Hello delay of the Hello message

Configuring PIM-SMv6 neighbor filtering

Configuring DR priority

Configuring static RP

Configure candidate BSR

Configure BSR border

Ignoring the RP priority of RP-SET

Configuring candidate RP

Checking the reachability of RP registration message

Filtering the addresses of registration packets on RP

Limiting the rate to send registration packets

Configuring the calculation method of checksum of registration packets

Limiting the range of legal BSRs

Configuring elected BSR to limit the address range of legal candidate RP and the multicast group range it serves

Enabling elected BSR to receive the candidate RP advertisement whose prefix-count is 0

Please configure this command to make elected BSR receive C-RP-ADV whose prefix-count is 0. If this command is not configured, the elected BSR cannot deal with C-RP-ADV whose prefix-count is 0.

Please run the following command in the global configuration mode.

Command	Function
ipv6 pim accept-crp-with-null-group	Elected BSR can receive C-RP-ADV whose prefix-count is 0.
no ipv6 pim accept-crp-with-null-group	Elected BSR cannot receive C-RP-ADV whose prefix-count is 0.

Enabling Multicast Routing

Multicast packets can be forwarded and enabling PIM-SMv6 makes sense only after multicast routing is enabled.

To enable or disable multicast routing, run the following command in the global configuration mode.

Command	Function
Ruijie(config)# **ipv6 multicast-routing**	Enable multicast routing globally.
Ruijie(config)# **no ipv6 multicast-routing**	Disable multicast routing globally.

Enabling PIM-SMv6

In order for one device to interact with other devices on PIM-SMv6 control messages, maintain and update the multicast routing table and forward multicast packets, PIM-SMv6 must be enabled on every interface.

To enable PIM-SMv6 on the interface, run the following commands in the interface configuration mode.

Command	Function
Ruijie(config-if)# **ipv6 pim sparse-mode**	Enable PIM-SMv6 on the interface.
Ruijie(config-if)# **no ipv6 pim sparse-mode**	Disable PIM-SMv6 on the interface.

⚠ Caution Enabling PIM-SMv6 brings into effect only after multicast routing is enabled globally.

Caution

If an interface is the tunnel interface, only IPv6 over IPv4 tunnel, IPv6 over IPv4 GRE tunnel, IPv6 over IPv6 tunnel and IPv6 over IPv6 GRE tunnel support IPv6 multicast. You can also enable multicast on tunnel interfaces not supporting multicast. However, the system neither prompts error messages nor receives or sends multicast packets.

Caution

Multicast tunnels can be set up only on Ethernet interfaces. They do not support embedded tunnels or QoS/ACL of multicast data.

Configuring the Hello Message Transmission Interval

After enabling PIM-SMv6, the interface sends the Hello message to the ones of adjacent devices at an interval. This transmission interval can be set as required.

To configure the Hello message transmission interval, run the following command in the interface configuration mode.

Command	Function
Ruijie(config-if)# **ipv6 pim query-interval** *interval-seconds*	Set the Hello message transmission interval. *interval-seconds*: in the range 1 to 65535 seconds
Ruijie(config-if)# **no ipv6 pim query-interval**	Restore the setting to the default value.

By default, the Hello message transmission interval is 30 seconds on the interface.

Note

When the Hello message transmission interval is updated every time, the Hello message hold time is updated 3.5 times of the Hello message transmission interval. If the Hello message transmission interval multiplying 3.5 is larger than 65535, the Hello message hold time is updated to 65535.

Configuring the Propagation Delay of the Hello Message

After the interface sends the Hello message, you can set the options of the Hello message. For LAN prune delay, the Propagation_Delay field is 500 ms by default.

To configure the propagation delay, run the following command in the interface configuration mode.

Command	Function
Ruijie(config-if)# **ipv6 pim propagation-delay** *interval-milliseconds*	Set the propagation delay. *interval-milliseconds:* in the range of 1-32767 ms.
Ruijie(config-if)# **no ipv6 pim propagation-delay**	Restore the propagation delay setting to the default value, namely 500 ms.

Note

Change of propagation delay or prune delay will influence the override interval of Join/prune message. As specified in the protocol, the override interval of Join/prune message must be less than its hold time or otherwise this will cause temporary interruption.

Configuring the Override Interval of the Hello Message

After the interface sends the Hello message, you can set the options of the Hello message. For LAN prune delay, the override-interval field is defaulted to 2500ms.

To configure the override-interval field, run the following command in the interface configuration mode.

Command	Function
Ruijie(config-if)# **ipv6 pim override-interval** *interval-milliseconds*	Set the override-interval. *interval-milliseconds:* In the range of 1 to 65535 seconds.
Ruijie(config-if)# **no ipv6 pim override-interval**	Restore the setting to the default value.

Note

Change of propagation delay or prune delay will influence the override interval of Join/prune message. As specified in the protocol, the override interval of Join/prune message must be less than its hold time or otherwise this will cause temporary interruption.

Configuring the Neighbor Tracking of the Hello Message

After an interface sends the Hello message, the LAN Prune Delay option of the Hello message has a T bit, indicating whether join constraint is enabled on the interface. With join constraint enabled, the interface is constrained not to send its Join message to the upstream neighbor when it receives the Join message that its neighbor sends to the upstream neighbor. On the other hand, with join constrain disabled, the interface will send its Join message to the upstream neighbor when it receives the Join message that its neighbor sends to the upstream neighbor. This function allows upstream routers to track how many receivers in downstream in accord with all received Join messages. Join constraint is enabled on the interface by default.

To disable join constraint on the interface, run the following command in the interface configuration mode.

Command	Function
Ruijie(config-if)# **ipv6 pim neighbor-tracking**	Disable join constraint on the interface.
Ruijie(config-if)# **no ipv6 pim neighbor-tracking**	Enable join constraint on the interface

Configuring the Triggered Hello Delay of the Hello Message

When the router starts or detects a new neighbor starts, it sends the Hello message. To avoid congestion, the router will send the Hello message at random. This random time is calculated by triggered hello delay, which is 5s by default.

To configure triggered hello delay, run the command in the interface configuration mode.

Command	Function
Ruijie(config-if)# **ipv6 pim triggered-hello-delay** *interval-seconds*	Configure triggered hello delay. *interval-seconds*: In the range of 1 to 5 seconds
Ruijie(config-if)# **no ipv6 pim triggered-hello-delay**	Restore the setting to the default value.

Configuring PIM-SMv6 Neighbor Filtering

Neighbor filtering can be enabled on the interface for security. PIM-SMv6 will not establish adjacency relation with a neighbor device and delete the established adjacency relation as long as the neighbor device is denied by the filtering access list.

To configure neighbor filtering, run the following command in the interface configuration mode.

Command	Function
Ruijie(config-if)# **ipv6 pim neighbor-filter** *ipv6_access-list*	Enable neighbor filtering on the interface.
Ruijie(config-if)# **no ipv6 pim neighbor-filter** *ipv6_access-list*	Disable neighbor filtering on the interface.

By default, neighbor filtering is disabled on the interface.

Note

Only the neighbors whose IP address matches ACL filtering can serve as the PIM neighbors of the interface. The neighbor addresses filtered by ACL cannot function as the PIM neighbors of the interface.

Configuring DR Priority

Use this command to set the priority of a device. The higher the value, the higher the priority.

Run the following commands in the interface mode:

Command	Function
Ruijie(config-if)# **ipv6 pim dr-priority** *priority-value*	Configure DR priority in the range of 0 to 4294967294.
Ruijie(config-if)# **no ipv6 pim dr-priority** *priority-value*	Restore DR priority to the default value, namely 1.

Configuring Static RP

In a smaller network, you can use PIM-SMv6 by configuring static RP. All devices in the PIM-SMv6 domain should be configured with similar static RP for consistent PIM-SMv6 multicast routes.

To configure static RP, run the following command in the global configuration mode.

Command	Function
Ruijie(config)# **ipv6 pim rp-address** *ipv6_rp-address* [*ipv6_access-list*]	Configure static RP.
Ruijie(config)# **no ipv6 pim rp-address** *ipv6_rp-address* [*ipv6_access-list*]	Remove the configuration.

Note that:

- When BSR and static RP take effect simultaneously, static RP is preferred.
- Static RP can be configured for many multicast groups (by using ACL) or all multicast groups. However, one static RP can be configured only once.
- If more than one IPv6 address is configured to be RP, the highest IPv6 address is adapted first.
- Only the IPv6 addresses defined and permitted by ACL are valid multicast groups. By default, all multicast groups are permitted.
- After configuration, the source address of static RP is inserted in the tree structure of the group range-based static RP group. Each group range-based static RP group maintains a chain structure that lists static RP groups in the descending order of IPv6 addresses. When a group range selects a RP, the first element, or the highest IPv6 address is selected.
- Deleting a static RP will delete it from all multicast groups and a new one is selected from the static RP tree structure as static RP.

Configuring Candidate BSR

Configuration of candidate BSR produces globally unique BSR in the PIM-SMv6 domain, which collects and distributes the RPs in the domain for the uniqueness of RP mapping.

To configure candidate BSR, run the following command in the global configuration mode.

Command	Function
Ruijie(config)# **ipv6 pim bsr-candidate** *interface-type interface-number* [*hash-mask-length*] [*priority-value*]	Set the device as the candidate BSR. *hash-mask-length:* In the range 0 to 128, 126 by default. *priority-value:* In the range 0 to 255, 64 by default.
Ruijie(config)# **no ipv6 pim bsr-candidate**	Remove the configuration.

Note

To set an interface as the candidate BSR, it must be configured with a global unicast address used for unicast routes or a local address. The first global unicast address or local address is elected as the candidate BSR.

Configuring BSR Border

To restrain BSM flooding, configure BSR border on the interface so that the interface drops BSM messages upon receiving them.

To configure BSR border, run the following command in the interface configuration mode.

Command	Function
Ruijie(config-if)# **ipv6 pim bsr-border**	Set the interface to be BSR border.
Ruijie(config-if)# **no ipv6 pim bsr-border**	Remove the configuration.

Ignoring the RP Priority of RP-Set

When you select a RP for a multicast address, ignore their priorities in comparison if there are more than one RP. If this command is not configured, the priority is used to compare two RPs.

To ignore RP priority, run the following command in the global configuration mode.

Command	Function
Ruijie(config)# **ipv6 pim ignore-rp-set-priority**	Ignore RP priority of RP-SET.
Ruijie(config)# **no ipv6 pim ignore-rp-set-priority**	Compare the RP priority of RP-SET.

Configuring Candidate RP

Configure candidate RP to periodically send candidate RP advertisement to the BSR so that the candidate RP advertisement is propagated to all PIM-SMv6 devices in the domain and guarantee the uniqueness of RP mapping.

To configure candidate RP, run the following command in the global configuration mode.

Command	Function
Ruijie(config)# **ipv6 pim rp-candidate** *interface-type interface-number* [**priority** *priority-value*] [**interval** *interval-seconds*] [**group-list** *ipv6_access-list*]	Configure candidate RP. *priority-value*: In the range 0 to 255, 192 by default *interval-seconds*: In the range 1 to 16383s, 60s by default *ipv6_access-list*: All multicast groups are allowed by default.
Ruijie(config)# **no ipv6 pim rp-candidate**	Remove the configuration.

Note

To set an interface as the candidate RP of the specific group range, use this command with ACL option. Note that the calculation of group range is only based on the ACE with permit rule, not deny rule.

Checking the Reachability of RP Registration Message

This command can be used to check whether the RP is reachable before the DR sends the registration message to the RP. In other words, the DR queries the unicast routing and static multicast routing tables to learn whether a route to the RP exists. If not, DR does not send the register packet.

To check the reachability of RP, run the following command in the global configuration mode.

Command	Function
Ruijie(config)# **ipv6 pim register-rp-reachability**	Check the reachability of the RP.
Ruijie(config)# **no ipv6 pim register-rp-reachability**	Remove the configuration.

Caution

If there is a static multicast route to the RP and the next hop of this static multicast route is reachable in the unicast routing table, PIM-SMv6 considers that the RP is reachable, even though the RP is not reachable in the unicast routing table.

Filtering the Addresses of Registration Packets on RP

This command filters the source addresses and group addresses of the registration packets arrived on RP. If this command is not configured, each register packet is allowed to reach the RP. Only the registration packets whose source addresses and group addresses are permitted by ACL or the route map are processed. Other registration packets are filtered and the Register-stop message is sent back.

Run the following command in global configuration mode.

Command	Function
Ruijie(config)# **ipv6 pim accept-register** {**list** *ipv6_access-list* | **route-map** *map-name* }	Filter the source addresses and group addresses of registration packets.
Ruijie(config)# **no ipv6 pim accept-register**	Remove the configuration.

Limiting the Rate to Send Registration Packets

This command applies to the registration packets in (S, G) state, not the overall system.

To limit the rate to send registration packets, run the following commands in the global configuration mode.

Command	Function
Ruijie(config)# **ipv6 pim register-rate-limit** *rate*	Set the maximum number of registration packets sent per second, in the range of 1 to 65535.
Ruijie(config)# **no ipv6 pim register-rate-limit**	Remove the configuration.

Configuring the Calculation Method of Checksum of Registration Packets

This command calculates the checksum of all the packets of PIM protocol, including encapsulated multicast packets.

Without this command, the checksum of registration packets is calculated by default methods of PIM protocol.

To configure the calculation method of checksum of registration packets, run the following command in the global configuration mode.

Command	Function
Ruijie(config)# **ipv6 pim register-checksum-wholepkt** [**group-list** *ipv6_access-list*]	Calculate the checksum for registering packets based on the overall packets. When the **group-list** *ipv6 access-list* value is the default value, apply the configuration to all multicast addresses.
Ruijie(config)# **no ipv6 pim register-checksum-wholepkt** [**group-list** *ipv6_access-list*]	Remove the checksum calculation for registering packets. When the **group-list** *ipv6 access-list* value is the default value, apply the configuration to all multicast addresses.

Note

Some vendors calculate checksum based on the overall registration packets. Ruijie Networks introduces this function for the compatibility with these vendors.

If the device of other vendors serves as the RP and the Ruijie device serves as the source DR, you can use this command on the source DR. If the device of other vendors serves as the source DR and the Ruijie device serves as the RP, you can use this command on the RP.

Limiting the Range of Legal BSRs

This command limits the range of legal BSRs. Without this command the PIM-SMv6 enabled router receives all external BSM packets.

To limit the range of legal BSRs, run the following command in the global configuration mode.

Command	Function
Ruijie(config)# **ipv6 pim accept-bsr list** *WORD*	Filter the BSM packets of BSRs.
Ruijie(config)# **no ipv6 pim accept-bsr list**	Remove the configuration.

Note

This command filters the BSR address field of BSM packets. If the address of a BSM packet is denied by ACL, the BSM packet is filtered.

Configuring Elected BSR to Limit the Address Range of Legal Candidate RP and the Multicast Group Range it Serves

This command configures elected BSR to limit the address range of legal candidate RP and the multicast group range it serves. Without this command, the elected BSR receives all external advertisements of candidate RPs.

For the ACL rule of this command, source specifies the address of candidate RP, and destination specifies the multicast group range that the candidate RP serves. If the ACL denies both addresses, the multicast group range of the candidate RP is filtered.

To configure elected BSR to limit the address range of legal candidate RP and the multicast group range it serves, run the following command in the global configuration mode.

Command	Function
Ruijie(config)# **ipv6 pim accept-crp list** *WORD*	Elected BSR filters the advertisement of candidate RP.
Ruijie(config)# **no ipv6 pim accept-crp**	Remove the configuration.

Enabling Elected BSR to Receive the Candidate RP Advertisement whose Prefix-count is 0

This command enables the elected BSR to receive the candidate RP advertisement whose prefix-count is 0. Without this command, the elected BSR will not process this kind of packets.

Once configured, the elected BSR considers that the candidate RP supports all multicast groups upon receiving the candidate RP advertisement whose prefix-count is 0.

To enable the elected BSR to receive the candidate RP advertisement whose prefix-count is 0, run this command in the global configuration mode.

Command	Function
ipv6 pim accept-crp-with-null-group	The elected BSR can receive the C-RP-ADV packet whose prefix-count is 0.
no ipv6 pim accept-crp-with-null-group	The elected BSR cannot receive the C-RP-ADV packet whose prefix-count is 0.

Configuring the Source Address of Registration Packets

This command configures the source address of registration packets. Without this command or with the no form of this command, the interface address of DR connecting to the multicast source is used. For address parameter of this command, the address to be set must be reachable to unicast routes. For interface parameter of this command, the interface to be set must be loopback interface or other type of interface whose address is advertised by unicast routes, in which the first non-local link address of the interface serves as the source address of registration packets.

To configure the source address of registration packets, run the following command in the global configuration mode.

Command	Function
Ruijie(config)# **ipv6 pim register-source** {*ipv6_local_address* | *interface-type interface-number*}	Configure the source address of registration packets.
Ruijie(config)# **no ipv6 pim register-source**	Use the RPF's interface address as the source address of registration packets.

Configuring the Suppression Time of Registration Packets

When the receiver does not receive the data packets destined to a multicast group from RP (namely RP does not serve this multicast group) or RP begins to receive multicast packets from the multicast source, RP sends the registration stop message to the DR on the multicast source side. Upon receiving this message, DR stops sending the registration packets encapsulated with multicast packets and transfers into the register suppression state.

During registration suppression, DR sends null registration packets, namely registration packets not encapsulated with multicast packets), to DR, indicating that the multicast source is still active. Probe time refers to the period that DR is allowed to send null registration packets before the registration suppression state is timed out. When registration suppression is timed out, DR starts to send registration packets. The smaller registration suppression timeout means the higher frequency that RP receives multicast packets; the larger timeout means the higher delay for a receiver to join a multicast group.

Running this command on DR will change the registration packet suppression time defined on DR.

To configure the registration packet suppression time, run this command in the global configuration mode.

Command	Function
Ruijie(config)# **ipv6 pim register-suppression** *seconds*	Configure registration packet suppression time. *Seconds*: in the range *1 to 65535*s
Ruijie(config)# **no ipv6 pim register-suppression**	Configure the suppression time to 60s.

Configuring the Probe Time of Null Registration Packet

The DR can send the null registration message to the RP in a period before the registration suppression time expires. This period is call probe time, 5 seconds by default

To configure the probe time of null registration packet, run the following command in the global configuration mode.

Command	Function
Ruijie(config)# **ipv6 pim probe-interval** *interval-seconds*	Configure the probe time of null registration packet. *interval-seconds*: in the range 1 to 65535 seconds
Ruijie(config)# **no ipv6 pim probe-interval**	Restore the probe time to 5s.

Note

The probe time must be less than half of registration suppression time. Furthermore, 3* registration suppression time plus registration probe time should be no more than 65535s or otherwise the system triggers an alarm.

Configuring RP KAT Timer

This command configures the hold time of (S, G) state that registration packets set up on RP.

To configure RP KAT timer, run this command in the global configuration mode.

Command	Function
Ruijie(config)# **ipv6 pim rp-register-kat** *seconds*	Configure the KAT timer, which ranges from 1 to 65535 seconds.
Ruijie(config)# **no ipv6 pim rp-register-kat**	Restore the setting to the default value, namely 3* registration suppression time plus registration probe time.

Caution

The timer should be larger than 3* registration suppression time plus registration probe time on the source DR, or otherwise the RP may time out the (S, G) state before the source DR sends the registration packet and thus leading to temporary interruption of multicast packets.

Configuring the Join/Prune Message Sending Interval

By default, the Join/Prune message is sent at the interval of 60s. You can use this command to change the sending interval.

To configure the Join/Prune message sending interval, run the following command in the global configuration mode.

Command	Function
Ruijie(config)# **ipv6 pim jp-timer** *interval-seconds*	Configure the Join/Prune message sending interval. *interval-seconds*: in the range 1 to 65535s

Ruijie(config)# **no ipv6 pim jp-timer**	Restore the setting to the default value, namely 60s.

Note

When you configure the Join/Prune message sending interval, if the sending interval * 3.5 is larger than 65535s, the system triggers an alarm and the sending interval is changed to be 65535/3.5 seconds.

Enabling the Last Hop Device to Transfer from the Shared Tree to the Shortest Path Tree

With this command, a PIM join message is triggered and a source tree is constructed upon the receipt of the first (S, G) message. The keyword **group-list** means all the groups in the list transfer to the source tree. After you use the **no** form of this command without the keyword **group-list**, to disable switching from the shared tree to the SPT, the device re-directs to the shared tree and sends a prune message to the source. If you use the previous command with a configured ACL, the association between the ACL and the **group-list** is removed, and all groups can switch from the shared tree to the SPT.

To enable the last hop device to transfer from the shared tree to the shortest path tee, run the following command in the global configuration mode.

Command	Function
Ruijie(config)# **ipv6 pim spt-threshold** [**group-list** *ipv6_access-list*]	If group-list is configured, the last hop device of this multicast group is enabled to transfer from the shared tree to the shortest path tree. Without group-list, all multicast groups are permitted.
Ruijie(config)# **no ipv6 pim spt-threshold** [**group-lis**t *ipv6_access-list*]	If the group-list is defined and the carried ACL is configured, the association between the ACL and the group-list is removed, and all multicast groups can switch from the shared tree to the shortest path tree. If group-list is not defined, the function of switching from the shared tree to the SPT is disabled.

Configuring the Specific Source Multicast

This command enables the device to directly receive multicast packets from the specific multicast source rather than the PR tree.

To configure the specific source multicast, run the following command in the global configuration mode..

Command	Function
Ruijie(config)# **ipv6 pim ssm** {**default** |**range** *ipv6_access-list*]}	Configure the specific source multicast, FF3x::/32 by default. x is an any valid range. If the ACL is specified, SSM is applied only to multicast groups that fall into this ACL.
Ruijie(config)# **no ipv6 pim ssm**	Remove the configuration.

Note

Interfaces on the last-hop switch or router connected to multicast receivers must enable MLDv2. Otherwise, the last-hop switch or router cannot process the MLD (S, G) Join message, and the specified source multicast will fail.

Configuring Static RP Preference

To configure the static RP's priority higher than the one elected through BSR mechanism, run the following command in the global configuration mode.

Command	Function
Ruijie(config)# **ipv6 pim static-rp-preferred**	Configure the static RP's priority higher than the one elected through BSR mechanism.
Ruijie(config)# **no ipv6 pim static-rp-preferred**	Configure the RP priority elected through BSR mechanism higher than the one of static RP.

Enabling Embedded RP

Embedded RP is the special RP discovery mechanism for IPv6 PIM that uses its IPv6 multicast address, from which the multicast router can directly resolve RP's address.

By default, embedded RP is enabled for the IPv6 multicast addresses of all embedded RP addresses. To enable embedded RP for the IPv6 multicast address of some embedded RP address, run this command in the global configuration mode.

Command	Function
Ruijie(config)# **ipv6 pim rp embedded** [**group-list** *ipv6_acl_name*]	Without *ipv6_acl_name,* enable embedded RP for the IPv6 multicast addresses of all embedded RP addresses. With *ipv6_acl_name,* enable embedded RP for the IPv6 multicast address of some embedded RP address.
Ruijie(config)# **no ipv6 pim rp embedded**	Disable embedded RP for the IPv6 multicast addresses of all embedded RP addresses. After embedded RP is disabled, the group address of the embedded RP is considered as an ordinary one, which means the RP mapping to the group address is no longer the embedded RP. It must calculate the embedded RP based on the learned RP mapping or the one configured statically.

Note

In addition to enabling embedded RP on the device, which is enabled by default, you also need to configure static RP on devices acting as embedded RP, or otherwise the device cannot serve as RP, even though its interface has the same address as embedded RP.

PIM-SMv6 Monitoring and Maintenance

PIM-SMv6 offers show commands to show the information on PIM-SMv6 interface, multicast group and multicast routing table.

Showing PIM-SMv6 Status

Use the following commands to show PIM-SMv6 status.

Command	Function
Ruijie# **show debugging**	Show debugging switches.
Ruijie# **show ipv6 pim sparse-mode bsr-router**	Show BSR details.
Ruijie# **show ipv6 pim sparse-mode interface** [i*nterface-type interface-number* [**detail**]]	Show PIM-SMv6 interface information.
Ruijie# **show ipv6 pim sparse-mode local-members** [*interface-type interface-number*]	Show local MLD information of PIM-SMv6 interface.
Ruijie# **show ipv6 pim sparse-mode mroute** {*ipv6_group_address* | *ipv6_source_address* | }	Show PIM-SMv6 multicast routing information.
Ruijie# **show ipv6 pim sparse-mode neighbor** [**detail**]	Show PIM-SMv6 neighbors.
Ruijie# **show ipv6 pim sparse-mode nexthop**	Show PIM-SMv6 next hop information from NSM.
Ruijie# **show ipv6 pim sparse-mode rp-hash** *ipv6_group-address*	Show the RP information of the specific multicast group address.
Ruijie# **show ipv6 pim sparse-mode rp mapping**	Show all RPs and the groups they serve.
Ruijie# **show ipv6 pim sparse-mode track**	Show the number of received and sent PIMv6 packets.

Deleting Internal PIM-SMv6 Messages

The following commands delete internal PIM-SMv6 messages.

Command	Function
Ruijie# **clear ipv6 mroute** {* | *group_address* [*source_address*] }	Clear multicast routing entries.
Ruijie# **clear ipv6 mroute statistics** {* | *group_address* [*source_address*] }	Clear multicast routing entry statistics.
Ruijie# **clear ipv6 pim sparse-mode bsr rp-set ***	Clear RP-SET.
Ruijie# **clear ipv6 pim sparse-mode track**	Reset the beginning time of statistics and reset PIMv6 packet counter.

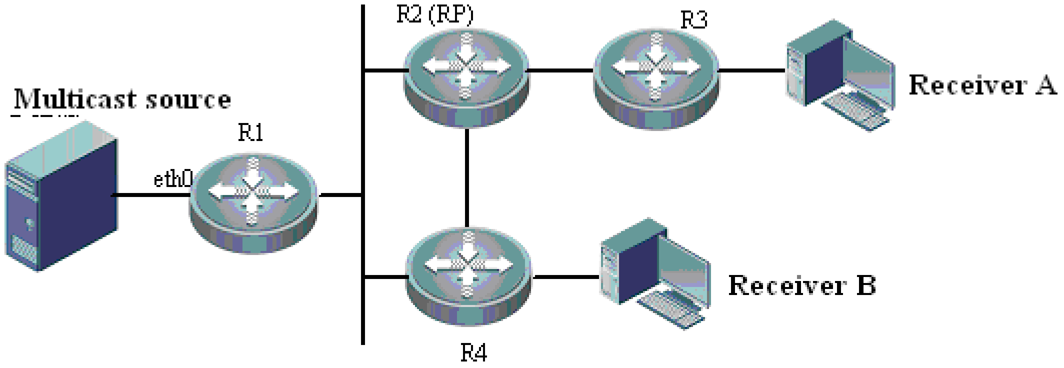
For details, refer to *PIM-SMv6 Command Reference*.

PIM-SMv6 Configuration Example

Configuration Requirements

Figure 3 illustrates network topology. R1 and the multicast source are located in one network. R2 is set to be RP. R3 and receiver A are located in the same network. R4 and receiver B are in the same network. Assume that devices are connected properly, IPv6 is enabled on each interface and IPv6 unicast is enabled on every device.

Figure 19 PIM-SMv6 topology



Configuration steps

Step1: Enable multicast routing

Enable IPv6 multicast routing on R1. The configuration is similar on R2, R3 and R4.

```
Ruijie# configure terminal
Ruijie(config)# ipv6 multicast-routing
```

Step 2: Enable PIM-SMv6 on the interface

Enable PIM-SMv6 on R1’s eth0. This configuration is similar on the interfaces of R1, R2, R3 and R4.

```
Ruijie(config)# interface eth 0
Ruijie(config-if)# ipv6 pim sparse-mode
Ruijie(config-if)# end
```

Step 3: Configure candidate BSR and candidate RP.

Set R2’s loopback1 to be C-BSR and C-RP

```
Ruijie(config)# interface loopback 1
Ruijie(config-if)# ipv6 address 2008:1::1/64
Ruijie(config-if)# ipv6 pim sparse-mode
Ruijie(config-if)# exit
Ruijie(config)# ipv6 pim bsr-candidate loopback 1
Ruijie(config)# ipv6 pim rp-candidate loopback 1
```

After the receiver joins the multicast group and the multicast source sends multicast packets, you can use show commands to monitor operation status.

Note MLD is automatically enabled on each interface while PIM-SMv6 is enabled.

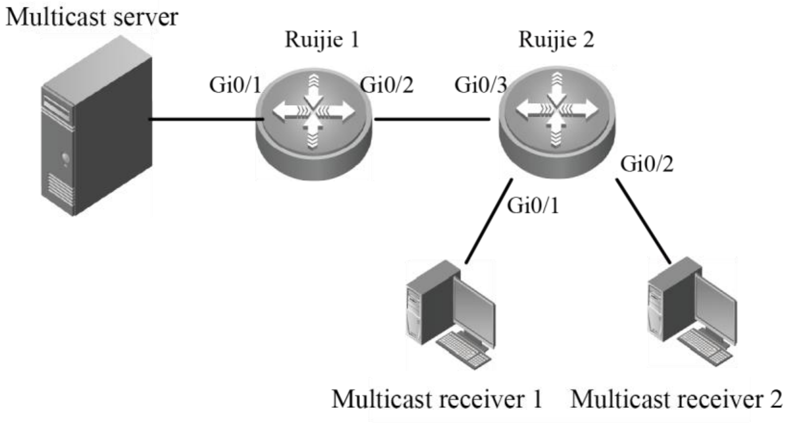
Configuration Examples of SSM Based on PIM

Networking Requirements

1. Network communication must be available at Layer 3. For example, run OSPFv3.
2. Multicast receivers obtain information about the multicast source through a channel depending on application software and deployment of network administrators on hosts.
3. Run PIM-SMv6 on the network.

Networking Topology

Figure 20 SSM configuration example



Basic configuration of devices and interfaces in Figure 4:

Device name	Interface	Basic configuration
Multicast server		Provides multicast data (2222::2, FF33::1111)
Ruijie 1	Gi 0/1	IPv6 address is 2222::1/64
Ruijie 1	Gi 0/2	IPv6 address is 3333::1/64
Ruijie 2	Gi 0/3	IPv6 address is 3333::2/64
Ruijie 2	Gi 0/1	IPv6 address is 4444::1/64
Ruijie 2	Gi 0/2	IPv6 address is 5555::1/64

Configuration Procedure

1. Basic configuration of interfaces on Ruijie 1 and Ruijie 2:

Configure interfaces on Ruijie 1 and Ruijie 2 according to the IPv6 addresses specified in the topology. The procedure is omitted.

2. Enable network interconnection on Layer 3:

Enable OSPFv3 on all interfaces of Ruijie 1 and Ruijie 2 and ensure devices are interconnected at Layer 3. The procedure is omitted.

3. Configure multicast on Ruijie 1 and Ruijie 2:

Enable the multicast routing on Ruijie 1 and Ruijie 2. Then, enable PIM-SMv6 on interfaces of Ruijie 1 and Ruijie 2. Lastly, enable MLDv2 on the interface connecting to multicast receivers of Ruijie 2. The following description uses Ruijie 2 as an example.

# Enable multicast routing on Ruijie 2:

```
Ruijie(conf)#ipv6 multicast-routing
```

# Enable PIM-SMv6 on all interfaces of Ruijie 2, such as interface Gi 0/1:

```
Ruijie(conf-GigabitEthernet0/1)#ipv6 pim sparse-mode
```

# Enable MLDv2 on the interface connected to the multicast receiver of Ruijie 2, such as interface Gi 0/1:

```
Ruijie(conf-GigabitEthernet0/1)#ipv6 mld version 2
```

The configuration steps of Ruijie 1 are similar to that of Ruijie 2. The procedure for configuring Ruijie 1 is omitted.

4. Enable SSM on Ruijie 1 and Ruijie 2:

# Enable SSM on Ruijie 2 and use the default SSM group range: FF3x::/32

```
Ruijie(conf)#ipv6 pim ssm default
```

# Enable SSM on Ruijie 1 and use the default SSM group range: FF3x::/32

```
Ruijie(conf)#ipv6 pim ssm default
```

Verifying the Configuration

1. Multicast receivers 1 and 2 request the multicast service (2222::2, FF33::1111) by sending MLD (2222::2, FF33::1111) join.

The protocol table of (2222::2, FF33::1111) will be created on Ruijie 2. You can view the protocol table by running the **show ipv6 pim sparse-mode mroute** command.

Ruijie#show ipv6 pim sparse-mode mroute

```
IPv6 Multicast Routing Table
(*,*,RP) Entries: 0
(*,G) Entries: 0
(S,G) Entries: 1
(S,G,rpt) Entries: 1
FCR Entries: 0
REG Entries: 0

(2222::2, FF33::1111)
RPF nbr: 3333::1
RPF idx: GigabitEthernet0/3
SPT bit: 0
Upstream State: JOINED
kat expires in 175 seconds
   00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Local
0.i   i  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
1  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
Joined
0  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
1  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
Asserted
0  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
1  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
Outgoing
0.o o  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
1  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
```

The output shows that the entry of (2222::2, FF33::1111) is created on Ruijie 2 and two multicast receivers exist.

The multicast server sends the data flow (2222::2, FF33::1111):

The multicast forwarding table of (2222::2, FF33::1111) will be created on Ruijie 2. You can view the multicast forwarding table with the **show ipv6 mroute** command.

```
Ruijie#show ipv6 mroute


IPv6 Multicast Routing Table

Flags: I - Immediate Stat, T - Timed Stat, F - Forwarder installed, R - RPT, S - SPT

Timers: Uptime/Stat Expiry

Interface State: Interface (TTL)


(2222::2, FF33::1111), uptime 00:19:31, stat expires 00:02:53

Owner PIMSMV6, Flags: TFSs

  Incoming interface: GigabitEthernet 0/3

  Outgoing interface list:

      GigabitEthernet 0/1(1)

      GigabitEthernet 0/2(1)
```

The output indicates that the multicast forwarding table of (2222::2, FF33::1111) is established on Ruijie 2 and the multicast egresses are Gi 0/1 and Gi 0/2, which are connected with multicast receiver 1 and multicast receiver 2 respectively. In addition, the Flags of this forwarding table is tagged with ‘s’, indicating that this forwarding table uses the SSM model.

Configuration Examples of Embedded RP

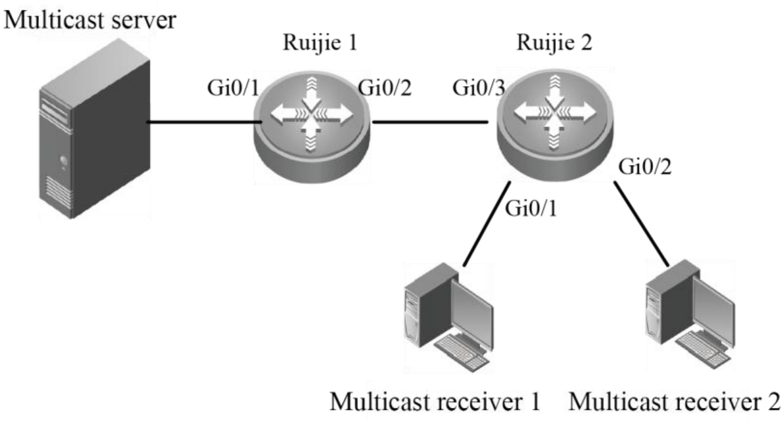
Networking Requirements

1. Network communication must be available at Layer 3. For example, run OSPFv3.

2. Run PIM-SMv6 on the network.

Networking Topology

Figure 21 Configuring embedded RP



Basic configuration of devices and interfaces in Figure 5:

Device name	Interface	Basic configuration
Multicast server		Provides multicast source (2222::2, FF7E:0540:6666::1)
Ruijie 1	Gi0/1	IPv6 address is 2222::1/64
Ruijie 1	Gi0/2	IPv6 address is 3333::1/64
Ruijie 2	Gi0/3	IPv6 address is 3333::2/64
Ruijie 2	Gi0/1	IPv6 address is 4444::1/64
Ruijie 2	Gi0/2	IPv6 address is 5555::1/64
Ruijie 2	Loopback0	IPv6 address is 6666::5/64

Configuration Procedure

1. Basic configuration of interfaces on Ruijie 1 and Ruijie 2:

Configure interfaces on Ruijie 1 and Ruijie 2 according to the IPv6 addresses specified in the topology. The procedure is omitted.

2. Configure the networks to interoperate on layer 3:

Enable OSPFv3 on all interfaces of Ruijie 1 and Ruijie 2 and ensure that networks are interoperable on layer 3. The procedure is omitted.

3. Configure the multicast on Ruijie 1 and Ruijie 2:

Enable multicast routing on Ruijie 1 and Ruijie 2, and then enable PIM-SMv6 on all interfaces on Ruijie 1 and Ruijie 2, and configure the static RP on Ruijie 2 to enable it acting as the embedded RP.

# Enable multicast routing on Ruijie 2:

```
Ruijie(conf)# ipv6 multicast-routing
```

# Enable PIM-SMv6 on all interfaces of Ruijie 2, with Gi 0/1 as an example:

```
Ruijie(conf-GigabitEthernet0/1)# ipv6 pim sparse-mode
```

# Configure static RP 6666::5 on Ruijie 2:

```
Ruijie(conf)# ipv6 pim rp-address 6666::5
```

The configuration steps of Ruijie 1 are similar with that of Ruijie 2; therefore, the procedure for configuring Ruijie 1 is omitted.

4. Enable embedded RP on Ruijie 1 and Ruijie 2:

By default, embedded RP is enabled. Enable it if it is disabled.

# Enable embedded RP on Ruijie 2:

```
Ruijie(conf)# ipv6 pim rp embedded
```

# Enable embedded RP on Ruijie 1:

```
Ruijie(conf)# ipv6 pim rp embedded
```

Verifying the Configuration

1. Multicast receivers 1 and 2 send MLD (*,FF7E:0540:6666::1) join:

The protocol table of (*,FF7E:0540:6666::1) is created on Ruijie 2. You can view the protocol table by running the show ipv6 pim sparse-mode mroute command. Meanwhile, the group RP mapping corresponding to the IPv6 multicast address FF7E:0540:6666::1 of the embedded RP will be saved on Ruijie 2. You can will the group RP mapping by running the show ipv6 pim sparse-mode rp mapping command.

```
Ruijie#sh ipv6 pim sparse-mode rp mapping

PIM Group-to-RP Mappings

Group(s): ff7e:540:6666::/96

  RP: 6666::5

    Info source: Embedded RP

      Uptime: 00:00:12, expires: 00:00:48
```

The preceding output indicates that on Ruijie 2, the group RP mapping of the IPv6 multicast address FF7E:0540:6666::/96 of the embedded RP is created automatically and the RP address is 6666::5.

```
Ruijie#show ipv pim sparse-mode mroute

IPv6 Multicast Routing Table
```

```
(*,*,RP) Entries: 0

(*,G) Entries: 1

(S,G) Entries: 0

(S,G,rpt) Entries: 0

FCR Entries: 0

REG Entries: 0


(*, ff7e:540:6666::1)

RP: 6666::5

RPF nbr: ::

RPF idx: None

Upstream State: JOINED

   00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

Local

0.i i  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .

1  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .

              Joined

0  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .

1  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .

Asserted

0  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .

1  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .

FCR:
```

The preceding display indicates that the protocol table of (*,ff7e:540:6666::1) has been created and two multicast receivers exist.

2. The multicast server sends the multicast data (2222::2, FF7E:0540:6666::1):

The multicast forwarding table of (2222::2, FF7E:0540:6666::1) will be created on Ruijie 2. You can view the forwarding table by running the show ipv6 mroute command.

```
Ruijie#show ipv6 mroute
```

```
IPv6 Multicast Routing Table

Flags: I - Immediate Stat, T - Timed Stat, F - Forwarder installed, R - RPT, S - SPT

Timers: Uptime/Stat Expiry

Interface State: Interface (TTL)


(2222::2, FF7E:0540:6666::1), uptime 00:19:31, stat expires 00:02:53

Owner PIMSMV6, Flags: TFS

  Incoming interface: GigabitEthernet 0/3

  Outgoing interface list:

      GigabitEthernet 0/1(1)

      GigabitEthernet 0/2(1)
```

Configuring RMEF

RMEF Overview

IP multicast realizes efficient point-to-multipoint data transmission over IP networks. Since IP multicast can effectively save network bandwidth and reduce network load, it is widely applied in real-time data transmission, multimedia conferencing, data copying, gaming and simulation.

Ruijie Multicast Express Forward (RMEF) maintains a mirror image of control-plane multicast routing table at the express forwarding data plane, so that forwarding of multicast packets can be done at the express forwarding data plane, thus improving the performance and efficiency of multicast forwarding.

Configuring REMF

Enabling/Disabling Multicast Express Forwarding On the Interface

Use the following commands to enable/disable multicast express forwarding on the interface in interface configuration mode.

Command	**Function**
Ruijie(config-if)# **ip ref**	Enables multicast express forwarding on the interface.
Ruijie(config-if)# **no ip ref**	Disables multicast express forwarding on the interface.

By default, multicast express forwarding is enabled on the interface.

Displaying RMEF Configuration and Status

Use the following commands to show RMEF configurations and statistics.

Command	**Function**
show ip ref mcast route [*ip address ip address*]	Shows the multicast express forwarding table. When no parameters are set, this command shows all table entries. When parameters are set (the first parameter is the IP address of the multicast source and the second one is the IP address of the multicast group), this command shows the table entries that match the two IP addresses.
show ip ref mcast info	Shows RMEF information, including the RMEF status, whether RMEF is enabled, number of express forwarding tables, and data packet uploading rate of RMEF in the case of no forwarding table or in data packet uploading process.
show ip ref mcast statistics **interface** *interface-type interface-number*	Shows statistics on the packets forwarded on the multicast express forwarding interface.

Command	Function
show ip ref mcast statistics mfc	Shows statistics on data packets forwarded on all multicast express forwarding entries.

Configuration Examples

The following examples show typical configurations on a router:

```
RSR20-04# config
```

! Enable multicast forwarding.

```
RSR20-04(config)# ip multicast-routing
```

! Enter the related interface and enable the multicast protocol.

```
RSR20-04(config)# interface fastEthernet 0/0
RSR20-04(config-if)# ip pim dense-mode
```

! Enable express forwarding on the interface.

```
RSR20-04(config-if)# ip ref
```

The following example shows how to debug multicast express forwarding:

! Display all multicast express forwarding tables.

```
Ruijie (config-if)# show ip ref mcast route
IP Multicast EF Routing Table
Interface State: Interface (Interface Index)
(30.1.1.2, 224.1.1.2)
 In_interface: GigabitEthernet 0/1.100(8)
 Hit: Yes
 To_cpu: No
 Oif_list: GigabitEthernet 0/2.100 (12)
```

! Display multicast express forwarding information.

```
Ruijie (config-if)# show ip ref mcast info
---------------------------------
IP RMEF is open
total RMEF MFC NUM = 1
to_cpu ratelimit PPS in one second = 10
no_mfc ratelimit PPS in one second = 10
---------------------------------
```

! Display multicast express forwarding statistics.

```
Ruijie (config-if)# show ip ref mcast statistics mfc
(30.1.1.2, 224.1.1.2)
 In_interface: GigabitEthernet 0/1.100(8)
 Match_PKTNUM: 17058555
```

```
Match_PKTBYTES: 1091747520
WRONG_IN_IF_PKTNUM: 0
TO_CPU_RESERVE_PACKET: 0
TO_CPU_DROP_PACKET: 0
Oif_list: GigabitEthernet 0/2.100(12)
```

Configuring MSDP

Overview

In PIM-SM network, network domains with same group-RP mapping information form a special multicast network domain, which is called PIM-SM domain.

1) The need for cross-domain multicasting

In the basic PIM-SM mode, a multicast source registers only with the RP in the local PIM-SM domain, and the multicast source information of a PIM-SM domain is isolated from that of another PIM-SM domain. As a result, the RP is aware of the source information only within the local domain and a multicast distribution tree is built only within the local domain; since RP is not aware of the multicast source information of other PIM-SM domains, it cannot join the multicast sources in other domains and build the multicast distribution tree. Therefore, cross-domain multicast application becomes impossible.

Given the feature of PIM-SM domain and the aforementioned problem, a mechanism is needed to allow RPs of different PIM-SM domains to share their multicast source information, so that the RP can learn the multicast source information of other PIM-SM domains. In this way, the local RP will be able to join multicast sources in other domains and the cross-domain multicast distribution tree can be built to carry out cross-domain multicast applications.

2) The need for Anycast-RP deployment

In a single PIM-SM domain, RP is the root node of share tree. RP is a critical device in PIM-SM domain, and it may fail due to excessive loads. Therefore, a RP load sharing mechanism will be needed to ease the burden of a single RP, and this mechanism is called Anycast-RP. In practice, we configure the same IP address for multiple RPs in a single PIM-SM domain; the multicast receiver joins the nearest RP according to the route, while the multicast source will register the multicast source information with the nearest RP according to the route, namely each RP will share certain PIM-SM activities.

During the deployment of Anycast-RP, we can encounter one problem: since the multicast source only registers the multicast source information with the nearest RP according to the route, other RPs acting as Anycast-RP will be unaware of the multicast source information and thus will not join the multicast source, leading to the failure of Anycast-RP deployment.

Given the aforementioned problem, a mechanism is needed to allow RPs acting as Anycast-RP to share their multicast source information, so that each RP acting as Anycast-RP can learn the multicast source information of the local domain. In this way, each RP will be able to join multicast sources and build the multicast distribution tree, allowing successful Anycast-RP deployment.

3) MSDP

Given the needs for cross-domain multicasting and Anycast-RP deployment, MSDP (Multicast Source Discovery Protocol) is introduced. With MSDP peer relationships established between different RPs, the RPs are interconnected with one another. If different RPs are distributed in different PIM-SM domains, then RP can discover the multicast source information in other PIM-SM domains, allowing cross-domain multicasting; if different RPs are distributed in the same PIM-SM domain, then RPs can share the multicast source information in the domain, allowing Anycast-RP deployment.

Basic Concepts

MSDP Peer

By establishing a special MSDP relationship between two layer-3 devices running MSDP protocol, these two layer-3 devices can share the multicast source information with each other. Such a special MSDP relationship is called MSDP peer relationship. Two layer-3 devices with MSDP peer relationship are called a pair of MSDP peers.

MSDP peer relationship is based on TCP connection and utilizes TCP 639 port. As long as the layer-3 route can reach, the MSDP peer relationship can be established between any two layer-3 devices.

Note The layer-3 device as mentioned herein refers to the generic router or layer-3 switch.

Peer-RPF Check of SA messages

MSDP device acting as RP will encapsulate the multicast source information registered with RP in SA messages, and forward SA messages to MSDP peers, which will then be able to learn the multicast source information.

Since the MSDP peers of the local MSDP device may form a loop. To avoid looping, MSDP protocol has introduced Peer-RPF check of SA messages.

The basic principle of Peer-RPF check of SA messages: According to the source address of SA messages, MSDP device will select the only MSDP peer for each SA message, and this MSDP peer is called Peer-RPF neighbor. When the local MSDP device receives SA message from Peer-RPF neighbor, it will accept this SA message and forward to other MSDP peers. Otherwise, it will drop this SA message.

Peer-RPF Forwarding of SA Messages

Once a MSDP device receives a SA message from the Peer-RPF neighbor, it will need to forward this SA message to other MSDP peers. This kind of behavior is called: Peer-RPF forwarding.

Mesh Group

An MSDP mesh group refers to a group of MSDP peers that have MSDP peering relationships among one another, namely any two layer-3 devices form a pair of MSDP peers.

When a SA messages is accepted by a MSDP peers in the mesh group, this SA message will not be forwarded to other MSDP peers in the same mesh group. This has reduced the flooding of SA messages and simplified Peer-RPF forwarding of SA messages.

Working Principle

1. MSDP peer

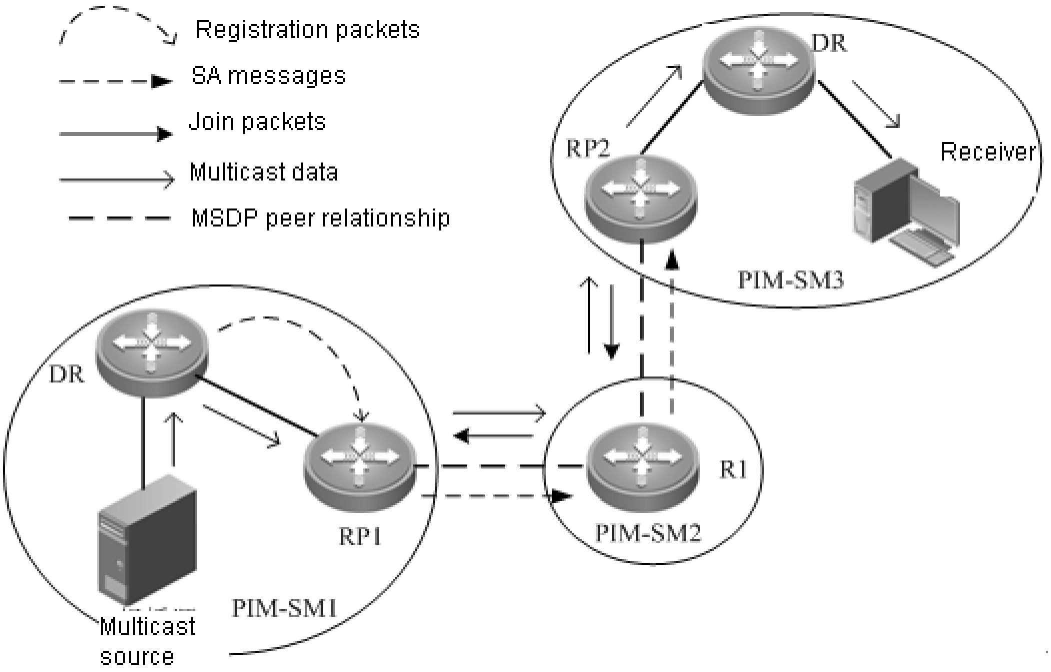
By configuring one or more pairs of MSDP peers in the multicast network, different RPs will be interconnected and send the multicast source information learned to MSDP peers, so as to advertise the multicast source information to other RPs.

If the aforementioned RPs are in different PIM-SM domains, then these PIM-SM domains can be interconnected. If the aforementioned RPs are in the same PIM-SM domain, then the multicast source information of the same PIM-SM domain will be shared among RPs.

2. Create MSDP peer

You can create MSDP peer on any layer-3 device. Devices that assume different multicast roles function differently. These peers can be classified into MSDP peers created on RP and MSDP peers created on non-RP.

Fig 1 MSDP Peers



(1) Creating MSDP peers on RPs

Source-side MSDP peer

According to PIM-SM protocol, the multicast source information is resisted with RP via registration message, indicating that the RP nearest to the multicast source (source-side RP) is aware of the multicast source information. The source-side RP creates SA messages according to the multicast source information registered and sends the messages to other RPs. Therefore, the MSDP peer must be created on the source-side RP; otherwise it will not be able to advertise the multicast source information.

As shown in Figure 1, the multicast source is in PIM-SM1, while RP1 is the RP nearest to the multicast source. The multicast source information is registered with RP1. Therefore, the peer must be created on RP1, so that RP1 can advertise the multicast source information. For example, the MSDP peer relationship is created between RP1 and R1 in PIM-SM2, so that the multicast source information can be sent to R1.

Receiver-side MSDP peer

If the receiver-side RP is unable to learn the multicast source information, it will not be able to join the multicast source and build the multicast distribution tree. Therefore, the MSDP peer must be created on the RP nearest to the receiver and being unaware of the multicast source information.

As shown in Figure 1, the receiver is in PIM-SM3, while RP2 is the nearest RP. Therefore, the peer must be created on RP2 in order to learn multicast source information from other PIM-SM domains. For example: MSDP peer relationship is established between RP2 and R1 in PIM-SM2; R1 sends the multicast source information learned from RP1 to RP2, so that RP2 can learn the same multicast source information. Then, RP2 joins the multicast source and build the multicast distribution tree from PIM-SM1 to PIM-SM3.

Intermediate MSDP Peer

The RP which is neither nearest to the multicast source nor nearest to the multicast receiver, functioning as a relay of multicast source information.

(2) MSDP created on non-RP: It will only forward the SA messages. As indicated in Figure 1, R1 in PIM-SM2 only forwards the SA messages to other MSDP peers. For example: R1 forwards SA messages to RP2.

Protocol Specification

Please refer to RFC3618.

Default Configurations

The following table describes the default configurations of MSDP.

Function	Default setting
Create MSDP peer	No MSDP peer is created.
Specify the default MSDP peer	No default MSDP peer is specified.
Configure MSDP peer description	No MSDP peer description is configured.
Configure SA request filtering	SA request messages are not filtered.
Configure mesh group	No MSDP peer will join the mesh group.
Change the originator address of SA messages	The originator address of SA messages is the address of RP.
Configure MD5 key for MSDP peer	MD5 encryption is not applied on the TCP connections between MSDP peers.
Configure redistribution filtering	All multicast source information registered with RP will be propagated to the MSDP peers.
Configure SA filtering	All SA messages from MSDP peers will be accepted; Each SA message received will be forwarded to the MSDP peer.
Configure the upper threshold of SA learning	No upper threshold for the SA information learned from MSDP peers.
Deactivate MSDP peer	MSDP peer is not deactivated.

Configure TTL upper threshold	TTL upper threshold is not configured. If the SA messages received carry the multicast data, then the SA messages forwarded to other MSDP peers will still carry the multicast data.

Configuring MSDP Peer

(Required) Creating MSDP peer

(Optional) Configuring MSDP peer description

(Optional) Configuring MD5 encryption for MSDP peer

Creating MSDP Peer

To enable MSDP, you need to configure MSDP peer. Execute the following command in the global configuration mode.

Command	Function
Ruijie(config)# **ip msdp peer** *peer-address* **connect-source** *interface-type interface-number*	Create MSDP peer The main address of the interface specified by the key word of "connect-source" will be used as the source address of MSDP TCP connection.

Configuring MSDP Peer Description

Each MSDP peer is an IP address with poor readability. In order to better distinguish different MSDP peers, you can configure text descriptions for MSDP peers.

To configure MSDP peer description, execute the following command in the global configuration mode:

Command	Function
Ruijie(config)# **ip msdp description** *peer-address text*	Configure text description for MSDP peer.

Configuring MD5 Encryption for MSDP Peer

The connection between MSDP peers is based on TCP connection. By default, the TCP connection will not use MD5 encryption, which is to say, TCP connection will not be subject to identity authentication. Under circumstances with high security needs, the identity authentication of TCP connections will be needed. To enable the identity authentication of TCP connections, execute the following commands in the configuration mode.

Command	Function
Ruijie(config)# **ip msdp password peer** *peer-address [encryption-type] string*	Enable MD5 encryption of the TCP connections between MSDP peers.

Controlling the Propagation of Multicast Source Information

(Optional) Redistribution filtering

(Optional) Filtering SA request messages

Redistribution Filtering

When the multicast source registers with RP, RP will learn the multicast source information and generate SA messages. By default, all registered multicast source information will be distributed by RP.

To control which registered sources can be distributed, execute the following commands in the global configuration mode.

Command	Function
Ruijie(config)# **ip msdp redistribute** [**list** *access-list-name*] [**route-map** *route-map-name*]	Only multicast source information passing the filtering rule will be propagated to the MSDP peers. 1) If the "list" or "route-map" is not specified, then no multicast source information (S, G) will be propagated. 2) If the "list" or "route-map" is specified, then only the multicast source information (S, G) meeting the access list or route map rule will be propagated to MSDP peers. 3) If the "list" and "route-map" is specified at the same time, then only the multicast source information (S, G) meeting both the access list and route map rule will be propagated to MSDP peers.

Filtering SA Request Messages

MSDP peer will request the multicast source information about a certain multicast group from local MSDP device by sending SA request messages. If the local device has the multicast source information about a certain multicast group, then the local MSDP device will need to send SA reply message to this MSDP peer, so as to inform the MSDP peer of the relevant multicast source information.

By default, the MSDP device will accept all SA request messages from the MSDP peer and give replies. However, you can configure to ignore all SA request messages, or only accept the SA request messages from certain multicast groups and ignore those from other multicast groups through the standard ACL.

To filter SA request messages, execute the following command in the global configuration mode:

Command	Function

Ruijie(config)# **ip msdp filter-sa-request** *peer-address* [**list** *access-list-name*]	Filter SA request messages from the specified MSDP peer, and only rely to SA request messages passing the filtering rule. 1) If the "list" is not specified, then all SA request messages from this MSDP peer will be rejected. 2) If the "list" is specified, then only SA request messages meeting ACL will be accepted.

Controlling the Forwarding of Multicast Source Information

(Optional) Using MSDP filter

(Optional) Using TTL to limit the multicast data carried in SA messages

Using MSDP Filter

After accepting a SA message, the MSDP device will need to forward the SA message to other MSDP peers. By default, MSDP device will forward the received SA messages to other MSDP peers without any change. However, in certain deployment environment, we need to limit the multicast source information that can be learned by other MSDP peers. In such a case, we will need to control which multicast source information can be forwarded to other MSDP peers.

To control which multicast source information that can be forwarded to MSDP peers, execute the following commands in the global configuration mode:

Mode	Function
Ruijie(config)# **ip msdp sa-filter out** *peer-address* [**list** *access-list-name*] [**route-map** *map-name*] [**rp-list** *rp-access-list-name*] [**rp-route-map** *rp-route-map-name*]	Configure SA message filtering rule for the specified MSDP peer. Only the multicast source information passing the filtering rule will be forwarded to MSDP peers. 1) If no key word is specified, SA messages will not be sent to the specified MSDP peer. 2) If one or more key words are specified, only multicast source information (S,G) meeting all rules corresponding to these key words will be forwarded the MSDP peer.

Using TTL to Limit the Multicast Data Carried in SA Messages

The SA messages can carry multicast data. In certain deployment environment, for efficiency related considerations, the TTL value of multicast data may be required to reach a certain threshold. If the TTL value of multicast data is less than the threshold, then the multicast data will be separated from SA messages and discarded. For example: By limiting the TTL value of multicast data packets to 8, the TTL of multicast data packets sent to MSDP peers shall be greater than or equal to 8.

To configure TTL threshold, execute the following commands in the global configuration mode.

Command	Function
Ruijie(config)# **ip msdp ttl-threshold** *peer-address ttl-value*	Limit the TTL value of multicast data carried in SA messages. When a SA message is received from one MSDP peer, if this SA message carries multicast data with TTL value less than the TTL threshold configured, then the multicast data will be separated from the SA message, and the SA message without multicast data will then be forwarded to other MSDP peers.

Controlling the Acceptance of Multicast Source Information

By default, MSDP device will accept all SA messages from the Peer-RPF neighbor. However, we can configure filtering rule to control which SA messages will be accepted by the local device.

There are several ways to control the multicast source information from MSDP peers:

- Filter all SA messages from a certain MSDP peer;
- Filter by using the specified access control list;
- Filter based on route map.

To control the multicast source information that can be accepted by the local device, apply the filter in the global configuration mode.

Command	Function
Ruijie(config)# **ip msdp sa-filter in** *peer-address* [**list** *access-list-name*] [**route-map** *route-map-name*] [**rp-list** *rp-access-list-name*] [**rp-route-map** *rp-route-map-name*]	Configure inbound SA message filtering rule for the specified MSDP peer. Only the multicast source information passing the filtering rule will be accepted. 1) If no key word is specified, SA messages from the specified MSDP peer will not rejected; 2) If one or more key words are specified, only multicast source information (S,G) meeting all rules corresponding to these key words will be accepted.

Configuring Default Peer

When the MSDP device receives an SA message, MSDP device will need to determine whether this SA message is from Peer-RPF neighbor; if not, the MSDP device will discard this SA message. To avoid the absence of Peer-RPF neighbor, you can configure the default peer, which is also a default Peer-RPF neighbor.

In the global configuration mode, execute the following command to specify the default MSDP peer:

Command	Function
Ruijie(config)# **ip msdp default-peer** *peer-address* [**prefix-list** *prefix-list-name*]	Specify the default MSDP peer. If "prefix-list" is not specified, then all SA messages from this default MSDP peer will be accepted; otherwise, only SA messages meeting prefix-list rule will be accepted.

Configuring MSDP Mesh Group

MSDP mesh group consists of a group of fully connected MSDP peers. The SA messages received by one MSDP peer in the mesh group will not be forwarded to other MSDP peers in the same mesh group. Therefore, it can reduce the flooding of SA messages and simplify Peer-RPF computation.

To create mesh group, execute the following command in the global configuration mode.

Command	Function
Ruijie(config)# **ip msdp mesh-group** *mesh-name peer-address*	Configure a MSDP mesh group and specify one MSDP peer to join this mesh group.

Deactivating MSDP Peer

If you don't want to activate a MSDP peer but deleting this MSDP peer will lose the relevant configurations, you can then deactivate this MSDP peer. After the MSDP peer is deactivated, only the TCP connection is terminated, and the configurations of this MSDP peer won't be deleted.

In the global configuration mode, execute the following command to deactivate the MSDP peer:

Command	Function
Ruijie(config)# **ip msdp shutdown** *peer-address*	Deactivate MSDP peer.

Configuring the Address of Originator other than RP

By default, the originator address of SA messages uses the address of RP in PIM-SM domain.

During the deployment of Anycast-RP, MSDP peer relationship will be established between RPs with the same IP address. When the multicast source registers with a RP nearest to the route, this RP will send SA messages to other RPs. By default, the originator address of such SA messages will be the address of RP. Upon receipt of such SA messages, other RPs will find out that the originator address is the local address, and the computation of Peer-RPF neighbor may fail. There are three solutions to the abovementioned problem:

- Configure mesh group
- Configure default MSDP peer
- Change the originator address of SA messages

To change the originator address of SA messages, execute the following command in the global configuration mode**.**

Command	Function

Ruijie(config)# **ip msdp originator-id** *interface-type interface-number*	Configure the originator address of SA messages to the main IP address of the specified interface.

Monitoring and Maintaining MSDP

Use the following commands to monitor MSDP.

Command	Function
Ruijie# **debug ip msdp peer** [*peer-address*]	Debug a specific peer or all peers, including: peer connection establishment/shutdown, MSDP protocol packets reception/forwarding, etc.
Ruijie# **show ip msdp count** [*as-number*]	Display the number of sources and groups indicated in the SA messages generated by each autonomous system.
Ruijie# **show ip msdp mesh-group**	Display the information of mesh group.
Ruijie# **show ip msdp peer** [*peer-address*]	Display the detailed information of a specific or all MSDP peers.
Ruijie# **show ip msdp rpf-peer** *ip-address*	Display the information of RPF-Peer with the specified address.
Ruijie# **show ip msdp sa-cache** [*group-address*|*source–address*] [*group-address*|*source–address*] [*as-number*]	Display the status of multicast source (S,G) learned from MSDP peer.
Ruijie**#** **show ip msdp sa-originated**	Display the SA originated multicast source (S, G).
Ruijie# **show ip msdp summary**	Display the brief information of MSDP peers.
Ruijie# **clear ip msdp peer** *peer-address*	Reset the TCP connection with a specific MSDP peer; reset all MSDP message counters.
Ruijie# **clear ip msdp statistics** [*peer-address*]	Clear the statistics of a specific or all MSDP peers.
Ruijie# **clear ip msdp sa-cache** [*group-address*]	Clear all SA cache entries or those of a specific group.

Typical MSDP Configuration Example

Cross-domain Multicasting

Networking Requirements

1) Three ISPs maintains autonomous system AS 100, AS 200 and AS 300 respectively. OSPF is adopted for the internal communication within each AS, while BGP is adopted for exchanging routing information between the autonomous systems.

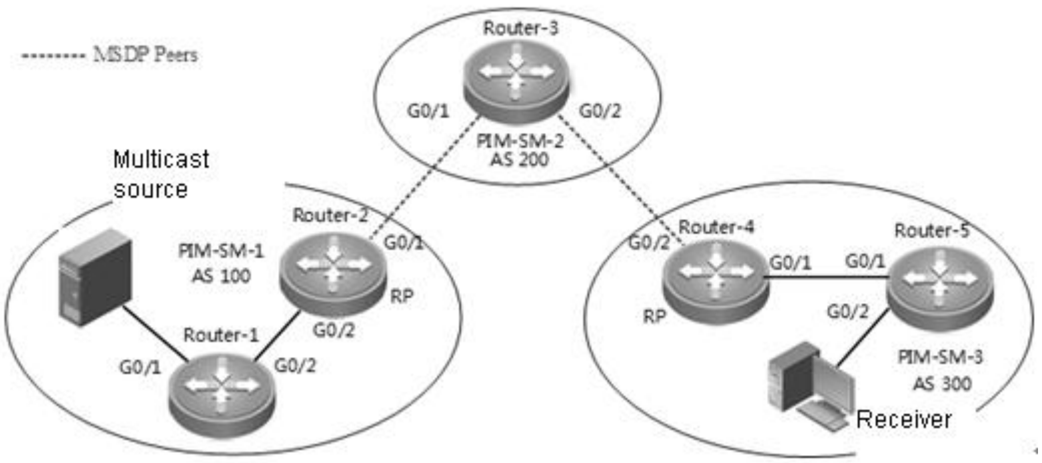
2) PIM-SM-1 belongs to AS 100, PIM-SM-2 belongs to AS 200, and PIM-SM-3 belongs to AS 300. Each PIM-SM domain has 0 or 1 multicast source and 0 or 1 receiver.

3) Configure the Loopback1 interface of Router-2 to C-BSR and C-RP of PIM-SM-1, and configure the Loopback1 interface of Router-4 to C-BSR and C-RP of PIM-SM-3.

4) Establish MSDP Peer relationship between Router-2 and Router-3 via EBGP, and establish MSDP Peer relationship between Router-3 and Router-4 via EBGP.

Network Topology

Fig 2 Cross-domain multicasting



Basic configurations of major interfaces in Figure 2.

```
Device      Interface       IP address
Router-1    G0/1            200.200.200.1/24
G0/2            100.100.100.1/24
Router-2    G0/2            100.100.100.2/24
G0/1            1.1.1.1/24
Loopback1       111.111.111.111/32
Loopback0       10.10.10.10/32
Router-3    G0/1            1.1.1.2/24
G0/2            2.2.2.1/24
Loopback0       30.30.30.30/32
Loopback1       40.40.40.40/32
Router-4    G0/2            2.2.2.2/24
G0/1            3.3.3.1/24
Loopback1       222.222.222.222/32
Loopback0       20.20.20.20/32
Router-5    G0/1            3.3.3.2/24
G0/2            4.4.4.1/24
```

Configuration Steps

(1) Configure the IP address and unicast routing protocol for the interfaces of respective devices

Configure the IP address and subnet mask of respective interfaces according to Figure 2. Configure OSPF protocol to connect the devices in each domain, and make sure the devices are interconnected at the network layer.

(2) Enable IP multicast routing, and enable PIM-SM on each interface.

# On Router-1, enable IP multicast routing, and enable PIM-SM on each interface.

```
Ruijie# conf t
Ruijie(config)# ip multicast-routing
Ruijie(config)# int g 0/1
Ruijie(config-if)# ip pim sparse-mode
```

Proceed with the similar configurations on other interfaces of the local device and on the interfaces of other devices.

(3) Configure BSR border

# Configure BSR border on G0/1 of Router-2.

```
Ruijie(config)# int g 0/1
Ruijie(config-if)# ip pim bsr-border
```

Configure BSR border on G0/1 and G0/2 of Router-3 and on G0/2 of Router-4.

(4) Configure C-BSR and C-RP

# Configure C-BSR and C-RP on Loopback1 of Router-2.

```
Ruijie# conf t
Ruijie(config)# int loopback 1
Ruijie(config-if)# ip pim sparse-mode
Ruijie(config-if)# ip address 111.111.111.111 255.255.255.255
Ruijie(config-if)# exit
Ruijie(config)# ip pim bsr-candidate loopback 1
Ruijie(config)# ip pim rp-candidate loopback 1
```

Configure Lookback1 of Router-4 to C-BSR and C-RP as shown above.

(5) Configure BGP routing protocol between the autonomous systems, and configure mutual route redistribution between BGP and OSPF

We need to configure EBGP Peer between Router-2 and Router-3 and between Router-3 and Router-4, and configure mutual route redistribution between BGP and OSPF. Please refer to BGP configuration guide for the detailed configuration process. You can execute BGP commands to display the route learning conditions between autonomous systems.

(6) Configure MSDP Peer

Configure Router-2 and Router-3 to MSDP peers.

# On Router-2, configure the main address of interface Loopback0 to the source address of TCP connection.

```
Ruijie(config)# ip msdp peer 30.30.30.30 connect-source loopback 0
```

# On Router-3, configure the main address of interface Loopback0 to the source address of TCP connection.

```
Ruijie(config)# ip msdp peer 10.10.10.10 connect-source loopback 0
```

Configure Router-3 and Router-4 to MSDP peers.

# On Router-4, configure the main address of interface Loopback0 to the source address of TCP connection.

```
Ruijie(config)# ip msdp peer 40.40.40.40 connect-source loopback 0
```

# On Router-3, configure the main address of interface Loopback1 to the source address of TCP connection.

```
Ruijie(config)# ip msdp peer 20.20.20.20 connect-source loopback 1
```

Verification

(1) Multicast source information

Multicast source sending multicast data (200.200.200.200, 225.1.1.1).

(2) Display MSDP peer status on Router-4

```
Ruijie#sh ip msdp summary
Msdp Peer Status Summary
Peer Address        As      State       Uptime/Downtime Reset-Count     SA-Count
    Peer-Name
40.40.40.40    200     Up          00:01:42           0                       1
    ?
```

(3) Display the multicast source information learned by Router-4

```
Ruijie#sh ip msdp sa-cache

MSDP Source-Active Cache: 1 entries
  (200.200.200.200,225.1.1.1),RP:111.111.111.111,MBGP/AS 100, 00:00:18/00:01:57, Peer
40.40.40.40
   Learned from peer 40.40.40.40, RPF peer 40.40.40.40,
  SAs received: 1, Encapsulated data received: 1
```

Deploying Anycast-RP

In Anycast-RP, two or more RPs with the same address are configured in the same PIM-SM domain, and MSDP peer relationship is established between these RPs, allowing load sharing and redundant backup between RPs in the domain.

The significance of Anycast-RP is shown below:

- RP with the shortest path: Multicast source registers with the nearest RP and build the SPT; the receiver joins the nearest RP and build the RPT.
- Load sharing between RPs: Each RP only maintains partial source/group information in the PIM-SM domain, allowing the load sharing between RPs.

- Redundant backup between RPs: When one RP fails, the multicast source registering with this RP or the receiver joining this RP will automatically select the nearest RP for registration or joining, allowing redundant backup between RPs.

Networking Requirements

In the PIM-SM domain, there are multiple multicast data senders and receivers; the OSPF protocol is running in the PIM-SM domain to allow intercommunication between devices at the network layer.

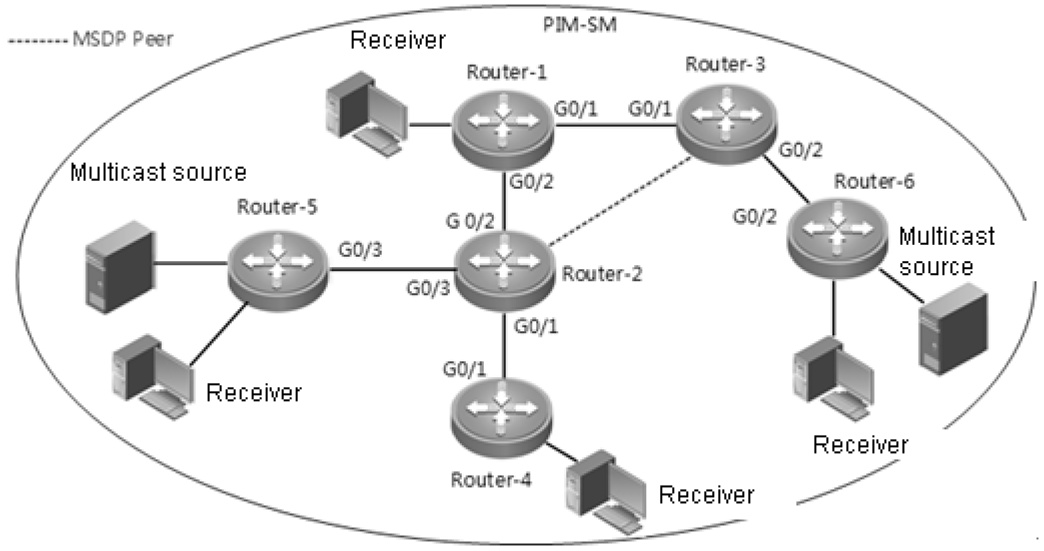
By configuring Anycast-RP in the PIM-SM domain, when a new member joins the multicast group, the receiver-side DR or router acting as DR can join the RP with the nearest unicast distance.

Establish MSDP Peer relationship between Router-2 and Router-3.

Configure C-BSR on Loopback0 of Router-1, and configure C-RP on Loopback1 of Router-2 and Router-3.

Network Topology

Fig 3 Anycast-RP deployment



Basic configurations of major interfaces in Figure 3.

Device	Interface	IP address
Router-1	G0/1	1.1.1.1/24
	G0/2	2.2.2.1/24
	Loopback0	100.100.100.100/32
Router-2	G0/1	3.3.3.1/24
	G0/2	2.2.2.2/24
	G0/3	4.4.4.1/24
	Loopback0	20.20.20.20/32
	Loopback1	10.10.10.10/32
Router-3	G0/1	1.1.1.2/24
	G0/2	5.5.5.1/24

	Loopback0	30.30.30.30/32
	Loopback1	10.10.10.10/32
Router-4	G0/1	3.3.3.2/24
Router-5	G0/3	4.4.4.2/24
Router-6	G0/2	5.5.5.2/24

Configuration Steps

(1) Configure the IP address and unicast routing protocol for the interfaces of respective routers

Configure the IP address and subnet mask of respective interfaces according to Figure 3. Configure to interconnect the routers via OSPF protocol.

(2) Enable IP multicast routing, and enable PIM-SM on each interface.

# On Router-1, enable IP multicast routing, and enable PIM-SM on each interface.

```
Ruijie# conf t
Ruijie(config)# ip multicast-routing
Ruijie(config)# int g 0/1
Ruijie(config-if)# ip pim sparse-mode
```

Proceed with the similar configurations on other interfaces of the local device and on the interfaces of other devices.

(3) Configure C-BSR and C-RP

# Configure C-BSR on Router-1

```
Ruijie(config)# int loopback 0
Ruijie(config-if)# ip pim sparse-mode
Ruijie(config-if)# ip address 100.100.100.100 255.255.255.255
Ruijie(config-if)# exit
Ruijie(config)# ip pim bsr-candidate loopback 0
```

# Configure Loopback1 of Router-2 and Loopback1 of Router-3 to the same address, and configure both interfaces to C-RP;

```
Ruijie# conf t
Ruijie(config)# int loopback 1
Ruijie(config-if)# ip pim sparse-mode
Ruijie(config-if)# ip address 10.10.10.10 255.255.255.255
Ruijie(config-if)# exit
Ruijie(config)# ip pim rp-candidate loopback 1
```

(4) Configure MSDP Peer

Configure Router-2 and Router-3 to MSDP peers.

# On Router-2, configure the main address of interface Loopback0 to the source address of connection.

```
Ruijie(config)#ip msdp peer 30.30.30.30 connect-source loopback 0
```

# On Router-3, configure the main address of interface Loopback0 to the source address of connection.

```
Ruijie(config)# ip msdp peer 20.20.20.20 connect-source loopback 0
```

Verification

(1) Multicast source information

There are two multicast source in Figure 2: (6.6.6.6, 225.1.1.1) and (7.7.7.7, 225.1.1.1). (6.6.6.6, 225.1.1.1) is registered with Router-2, while (7.7.7.7, 225.1.1.1) is registered with Router-3.

(2) Display MSDP peer status on Router-2

```
Ruijie#sh ip msdp summary
Msdp Peer Status Summary
Peer Address        As      State       Uptime/Downtime Reset-Count     SA-Count
    Peer-Name
30.30.30.30    ?       Up          00:01:42           0                       1
    ?
```

(3) Display the multicast source information learned by Router-2

The following messages show that Router-2 has learned the multicast source information (7.7.7.7, 225.1.1.1) registered with Router-3.

```
Ruijie#sh ip msdp sa-cache

MSDP Source-Active Cache: 1 entries
  (7.7.7.7,225.1.1.1),RP:10.10.10.10,MBGP/AS ?, 00:00:18/00:01:57, Peer 30.30.30.30
   Learned from peer 30.30.30.30, RPF peer 30.30.30.30,
SAs received: 1, Encapsulated data received: 1
```

Configuring Multicast VPN

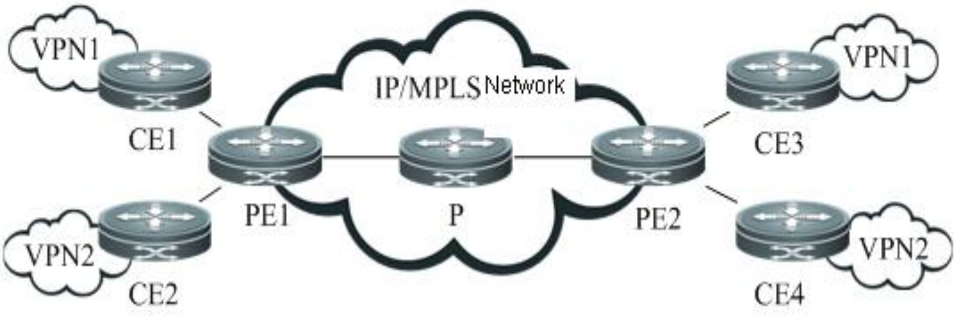
Introduction to multicast VPN

Overview

IP multicast is being widely applied, and it is being used as the solution to many applications in many sectors. Meanwhile, the application of VPN technology is getting more and more popular in the enterprise networks. BGP/MPLS L3VPN based architecture can be widely found in E-government network, electric power data network and other enterprise networks. By setting different departments into different VPNs, the data of different departments can be well isolated. Meanwhile, the intra-department video conferencing, data sharing and other multicast services will also involve VPN-based data isolation, leading to the increasingly pressing needs for multicast VPN.

Multicast VPN is a BGP/MPLS L3VPN network based technology to transmit multicast data within the VPN network, with network structure similar to that of BGP/MPLS L3VPN, as shown below:

Figure 22 Basic structure of multicast VPN



- CE

CE (Customer Edge Router) is an edge device of customer network, and is logically a customer VPN, with certain interface directly connected to the ISP device. CE can be a host, router or switch, such as CE1, CE2, CE3 and CE4 shown above.

- PE

PE (Provider Edge Router) is an edge device of SP backbone network (it can be a router, ATM switch, frame relay switch and etc), such as PE1 and PE2 as shown in the figure. Logically belonging to the service provider, PE is directly connected with CE, and one PE can be connected with multiple CEs.

- P

P (Provider Router) is the core device within the SP backbone network, such as P1, P2 and P3 as shown in the figure. P is not connected with CE, and is responsible for routing and fast forwarding on the public network.

As shown above, the multicast protocol is run on customer VPN to establish inter-VPN multicast routes and forward multicast traffic for the private network. When PE1 receives the multicast traffic from CE1 and CE2, it will convert the traffic into multicast data format that can be transmitted over the public network, and forward the traffic to PE2 of other VPN sites. PE2 will then de-encapsulate the traffic and forward to the corresponding VPN site.

Multicast Domain

MD (Multicast-Domain) is a scheme to realize multicast VPN, and is achieved by maintaining a multicast domain for each VPN in the service provider network. MD scheme will create a MDT (Multicast Distribution Tree) for reach VPN in the public network to transmit private network multicast protocol packets and data packets, which will be converted into public network multicast data packets on PE and be forwarded via MDT. There are two types of MDTs in the MD scheme:

➢ Default-MDT

Each multicast domain will create a Default-MDT to connect all PEs in the domain. Through this Default-MDT, the customer network can treat the multicast domain as a LAN, which is to say the CE multicast data transmitted to the ingress PE will be forwarded over the public network to all egress PEs in this multicast domain, while such multicast data are transmitted on Default-MDT in the form of multicast. The egress PE will determine whether or not to forward these multicast data by verifying whether there is multicast receiver in the VPN sites connected. If no receiver exists in the sites, the multicast data will be discarded.

➢ Data-MDT

Data-MDT is an optional optimized means to transmit multicast traffic. It will only forward to multicast traffic to the egress PE in need of such multicast traffic. When the ingress PE intends to create a Data-MDT, it will first send a MDT-join packet through Default-MDT, and such packets carries such information as C-S (source address of customer multicast traffic), C-G (group address of customer multicast traffic) and P-G (group address for transmission over public network), indicating that C-S and C-G identified multicast traffic will be transmitted along P-G identified Data-MDT. Upon receipt of MDT-Join packets, the egress PE will only join the Data-MDT when it is in need of such multicast traffic. Therefore, the multicast traffic will arrive at the egress PE in need of the multicast traffic along Data-MDT, thus saving the resources of irrelevant egress PEs and the bandwidth of public network. Hence, if the private network contains certain high-rate multicast traffic, we can forward such multicast traffic by configuring Data-MDT, so as to avoid bandwidth wasting caused by the use of Default-MDT.

Single-AS Multicast VPN

In the single-AS multicast VPN, we can use PIM-SM/SSM to create Default-MDT in order to interconnect different PEs.

➢ Use PIM-SM to create Default-MDT

In this scheme, by running PIM-SM on the public network, each PE uses the same group address (group address of Default-MDT) to join the RP on the public network, thus forming a shared tree (RPT) connecting RP and each PE. Then, each PE will register with RP, so that RP creates the SPT reaching each PE. In this way, the multicast packets sent by each PE can be transmitted through such SPT and RPT, and eventually reach other PEs. The Default-MDT is hence successfully created.

➢ Use PIM-SSM to create Default-MDT

In this scheme, by running PIM-SSM on the public network, each PE uses the MDT address-family routes of BGP to inform other PEs of its NLRI information (including PE address and Default-MDT group address). When one PE receives the MDT address-family route of BGP, it will use the PE address and Default-MDT group address contained therein to join SPT with the PE providing this route. A SPT is eventually formed with PE initiating the MDT address-family routes being the root and PEs receiving such MDT address-family routes being the leaves. Since each PE will initiate the MDT address-family route, multiple SPTs will be formed eventually. In this way, the multicast packets sent by each PE can be transmitted to other PEs via these SPTs, and Default-MDT is created accordingly.

Multi-AS Multicast VPN

When VPN needs to cross multiple ASes, there are three unicast solutions to address the PE communication problem between ASes. There are also different multicast VPN options based on these three unicast solutions.

- OptionA: VRF-to-VRF

VRF is configured for each VPN on the ASBR, which act as PE, and ASBRs are interconnected through their respective VPN instance and regard each other as a CE device. In this approach, ASBR acts as PE and the MDT is established within each AS. The multicast data is transmitted over the public network through MDTs of multiple ASes.

- OptionB: Single-hop EBGP

PE uses to IBGP to advertise the VPN route to the ASBR of local AS, and then ASBRs of different ASes will advertise the VPN route of other ASes to the local ASes through EBGP. In this way, ASBR no long acts as PE, and the MDT of multicast VPN must cross each AS in order to connect PEs of respective ASes.

- OptionC: Multi-hop EBGP

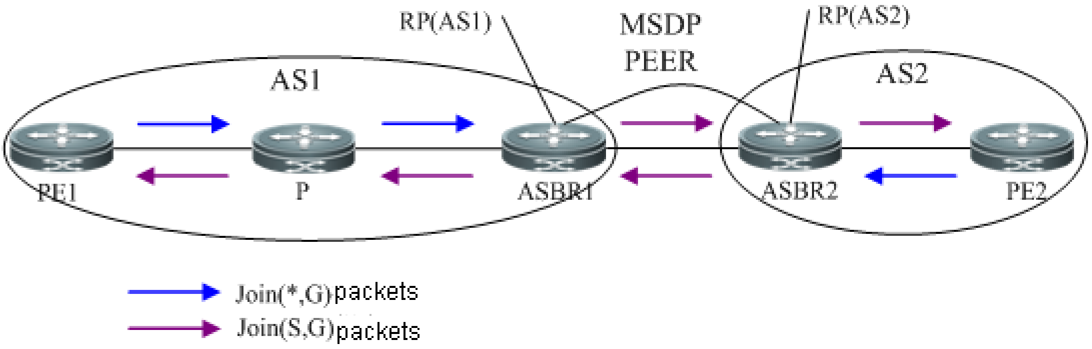
In this way, PEs of different ASes are interconnected via Multihop-EBGP, and ASBR will neither participate in the reception and advertisement of VPN route nor act as PE. In this approach, the MDT of multicast VPN must cross each AS in order to connect PEs of respective ASes.

In OptionA, since MDT is established within AS, the process is similar to that of single-AS multicast VPN. In OptionB and OptionC, when MDT is created using PIM-SM and PIM-SSM, there will be two different scenarios:

- Use PIM-SM to create multi-AS MDT

In order to interconnect different PEs in respective ASes via the same MDT, we can use MSDP to share multicast source information between RPs of two ASes. The process is shown below:

Figure 23 Use PIM-SM to create multi-AS MDT

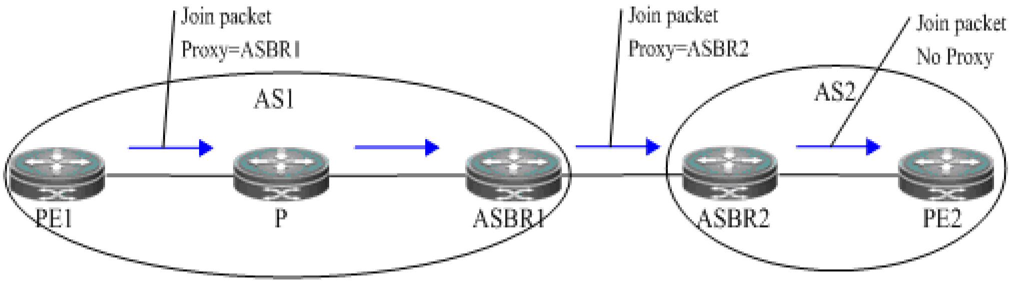


In this scenario, the ASBR of each AS is configured to MSDP peer, which is used to learn the multicast source information of other ASes, so as to initiate the grafting of SPT to other ASes. As shown above, PE1 and PE2 first initiate shared tree grafting to local RPs in order to create Default-MDT. When RP (ASBR1) in AS1 learns the multicast source information (PE2) of AS2 via MSDP, it will send Join (S,G) packets and eventually create a SPT with ASBR1 as the leave and PE2 as the root. Likewise, when ASBR2 learns the multicast source information (PE1) of AS1, a SPT with ASBR2 as the leave and PE1 as the root will also be created. In this way, the multicast packets sent by PE1 can be forwarded to ASBR2 along SPT, while ASBR2 will forward such packets to PE2 along the shared tree. Therefore, through the shared tree and SPT tree, PEs in different ASes can then be interconnected.

➢ Use PIM-SSM to create multi-AS MDT

In the process of using PIM-SSM to create multi-AS SPT, the P router in one AS may not have the route to the PE in another AS, and this will lead to the failure in SPT creation. To address this problem, PIM-SSM extends the attribute of Join packet so that it carries proxy information, while P router will use such proxy information to create SPT, as shown below:

Figure 24 Create multi-AS SPT



In the above figure, when PE1 initiates to create SPT, it will select ASBR1 as the proxy and include its IP address in the Join packet. When P receives this Join packet, it will use proxy as the destination address to discover RPF neighbors and initiate SPT grating to neighbor ASBR1 that leads to the proxy. After ASBR1 receives the Join packet, it will find out that the proxy address is its own IP address, and it will reselect a new proxy (ASBR2) and initiate SPT grafting. In this way, a SPT is formed through proxy grafting, so that PEs in respective ASes can be interconnected.

Note

When using PIM-SM to create multi-AS MDT, all routers along the path from RPs in respective ASes to the multicast source (namely PEs in other ASes) must have the unicast route to the multicast source. Therefore, this scenario requires advertising the route of each PE to the IGP of each AS.

Note

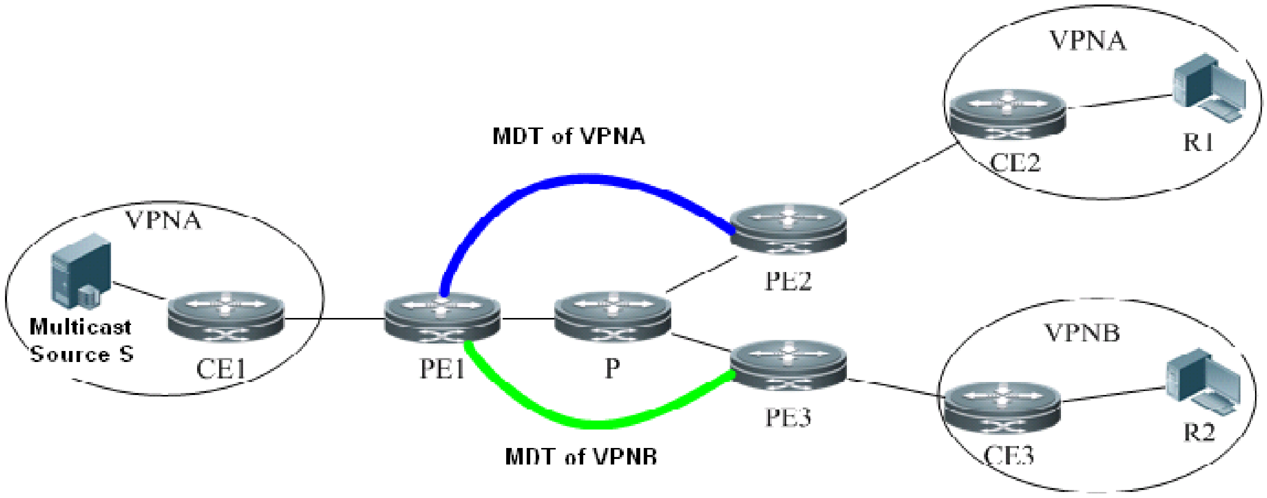
When using PIM-SSM to create multi-AS MDT, the P router is not required to have the route to PEs in other ASes. This scenario boasts higher flexibility, but will require PEs on the public network and P router to support proxy-based SPT creation.

Extranet Application of Multicast VPN

Extranet multicast VPN (Extranet-MVPN) enables the multicast source of a multicast VPN to send multicast streams to other multicast VPN sites. On the Extranet-MVPN, a cross-VPN sender is called an "Extranet-Source". The VRF where the sender is located is called the source MVRF. The PE connected to the source MVRF is called the ingress PE. A cross-VPN receiver is called an Extranet-Receiver. The VRF where the receiver is located is called a receiver MVRF. The PE connected to the receiver MVRF is called an egress PE. See the following figure. The multicast source on VPN A can forward the multicast stream to VPN B, so that the receiver on VPN B can receive the multicast stream. In this scenario, the VRF (on PE1) connected to the VPNA can be referred to as the source MVRF. Multicast source S can be referred to as the Extranet-Source. The VRF (on PE3) connected to VPN B can be referred to as the receiver MVRF. The receiver R2 can be referred to as the Extranet-Receiver.

On the Extranet-MVPN, a PE establishes a cross-VRF multicast distribution tree for introducing multicast streams of other VPN sites to the PE. Meanwhile, the PE uses the cross-VRF multicast stream forwarding function to forward streams from one VPN to another. The following figure shows the detailed process.

Figure 25 Work process of the Extranet-MVPN



As shown in the preceding figure, VPN A works as the sharing VPN; VPN B works as the receiver VPN; The VPNA unicast route is imported for VPN B on PE1 and PE3, so that CE3 can locate the route leading to the multicast source S. The corresponding VRF is configured for every VPN on all PEs and the MD function is enabled. They form the Extranet-MVPN for R2 between the multicast source S and VPN B.

➢ Establishment Process of MDTs on a Public Network

As shown in the preceding figure, the corresponding VRF is configured for every VPN on all PEs and the MD function is enabled (the VRFs corresponding to VPN A and VPN B are both configured on PE1). In this case, an MDT is created to connect PE1 VPN A to PE2 VPN A for transmitting multicast streams of VPN A on the public network; another MDT is created to connect PE1 VPN B to PE2 VPN B for transmitting multicast streams of VPN B on the public network.

➢ Process of Creating MDTs

The processes of creating MDTs are similar for PIM-SM and PIM-SSM. The only difference lies in that a multicast router initiates the graft process in the RP direction for PIM-SM, while a multicast router initiates the graft process in the multicast source direction for PIM-SSM. The process of creating an MDT for PIM-SSM is set out as follows as an example.

1) During the transmission from R1 to S, CE2 sends a Join packet to PE2 to create a MDT, and then PE2 uses the MDT of VPN A to send the Join packet to PE1. Upon receiving the Join packet, PE1 initiates the graft process to CE1, and eventually an MDT from CE1 to CE2 is created on VPN A.

2) During the transmission from R2 to S, CE3 sends a Join packet to PE3 to create an MDT, and then PE3 uses the MDT of VPN B to send the Join packet to PE1. Upon receiving the Join packet, PE1 performs the RPF check and finds that the RPF interface belongs to VRF A. PE1 notifies the PIM instance on VRF A of the receiving request. Then, the PIM instance on VRF A takes over the initiating of the MDT creation process. The PIM instance on VRF A sends the Join packet to CE1 for grafting. Finally, an MDT from CE1 (on VPN A) to CE3 (on VPN B) is created.

- Multicast Stream Forwarding Process

1) After a multicast stream of the multicast source S is forwarded through CE1 to PE1, PE1 forwards the multicast stream to PE2 and eventually to R1 through the MDT of VPN A.

2) After a multicast stream of the multicast source S is forwarded through CE1 to PE1, PE1 can detect VRF B's request on the multicast stream because the PIM instance of VRF B has notified the PIM instance of VRF A of the multicast-receving request when the MDT is created previously. In this case, PE 1 forwards the multicast stream to VRF B, transmits the multicast stream on the public network through the MDT on VPN B first to PE3 and eventually to R2.

Note

If a private network runs PIM-SM, the multicast source and RP must be on a same VPN. Otherwise, the source DR fails to register with RP and an MDT fails to be created.

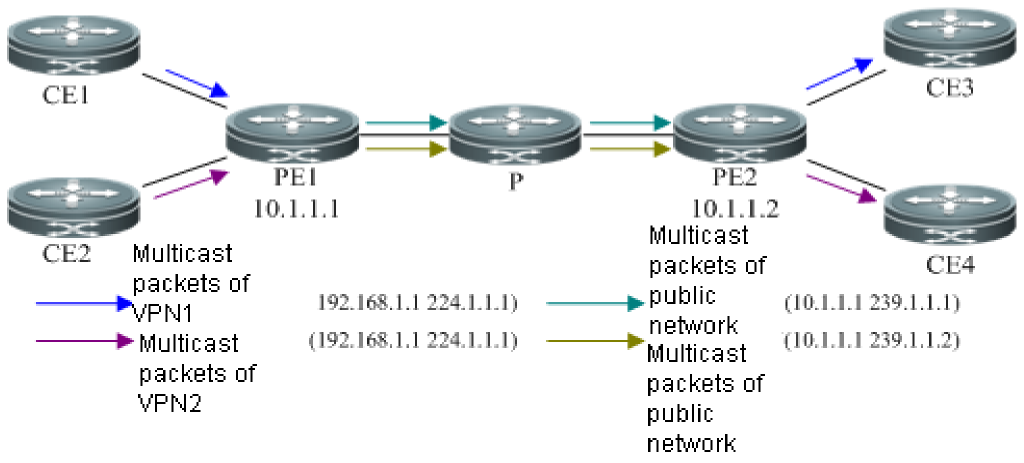
Multicast VRF

One PE may be connected with multiple CEs of different VPN sites. In order to differentiate the multicast traffic from different VPN sites on PE, the MD scheme also uses multiple VRFs to forward the multicast traffic from multiple VPN sites, and these VRFs are called MVRF (Multicast VPN Routing and Forwarding). The connection between PE and the CE of each VPN site needs to be associated with one VRF. When PE receives the multicast packets from CE, it will look up the multicast forwarding table under corresponding MVRF and forward packets as per such forwarding table. In this way, MVRF addresses the problem of local multicast route conflict on PE and isolate the multicast traffic in each VPN site.

The creation and maintenance process of MVRF is similar to that of VRF, yet multicast route must be initiated for each MVRF so that it can generate relevant multicast forwarding table. In the MD scheme, an MVRF is created on PE for each VPN to forward the corresponding multicast traffic. Meanwhile, each MVRF contains one Multicast Tunnel Interface (MTI), which is a GRE Tunnel interface converting private network multicast traffic into public network multicast traffic, which will then be transmitted over the public network through the public network multicast forwarding table.

Through the forwarding of private network multicast traffic by MVRF and the conversion of multicast traffic by MTI interface, the multicast data packets will eventually be forwarded from one VPN site to another VPN site over the public network. The packet forwarding process is shown below:

Figure 26 Forwarding process of multicast data packets


CE1
CE3
PE1
10.1.1.1
P
PE2
10.1.1.2
CE2
CE4
Multicast packets of VPN1
Multicast packets of VPN2
192.168.1.1 224.1.1.1)
(192.168.1.1 224.1.1.1)
Multicast packets of public network
Multicast packets of public network
(10.1.1.1 239.1.1.1)
(10.1.1.1 239.1.1.2)

As shown above, the forwarding process of multicast data packets is shown below:

1) CE1 and CE2 forward the multicast data packets in respective VPNs to PE1. The source address and destination address carried by data packets of two VPNs could be identical (as shown above, they are both 192.168.1.1, 224.1.1.1).

2) After PE1 receives the data packets transmitted from CE1 and CE2, it will select different MVRFs to forward these two multicast traffic according to the VPN. When each multicast traffic is forwarded to the MTI of respective MVRF, the multicast traffic will be encapsulated into the corresponding public network multicast traffic. As shown above, the multicast traffic of VPN1 is encapsulated into (10.1.1.1, 239.1.1.1), and that of VPN2 is encapsulated into (10.1.1.1, 239.1.1.2). These two multicast traffic will be transmitted to PE2 through public network multicast routing and forwarding.

3) PE2 will de-encapsulate these two public network multicast traffic and restore to the private network multicast traffic, which will then be forwarded by the MVRFs corresponding to their own VPNs and reach CE3 and CE4 respectively. Such traffic will eventually reach the multicast receivers in respective VPNs.

Caution To enhance the efficiency of packet processing, no GRE checksum will be calculated or configured during packet encapsulation on the MTI interface of ingress PE.

Caution Before packet encapsulation, the MTI interface of ingress PE cannot proceed with fragmentation first according to IP MTU of MTI interface.

Configuring multicast VPN

Configuring Single-AS Multicast VPN

Before configuring single-AS multicast VPN, the following network configurations must be completed first:

- Configure a unicast routing protocol to enable routes in AS
- Configure BGP/MPLS L3VPN

The configuration of single-AS multicast VPN mainly involves the following:

- Configure Default-MDT (Required)
- Configure Data-MDT (Optional)
- Configure multicast routing on VRF (Required)
- Configure PIM-SSM (Optional)
- Configure BGP MDT address family (optional)

Configuring Default-MDT

Steps of Default-MDT configuration are shown below:

Command	Function
Ruijie# **configure terminal**	Enter global configuration mode.
Ruijie(config)# **ip vrf** *vrf_name*	Create a VRF and enter VRF configuration mode.
Ruijie(config-vrf)# **rd** *route_distinguisher*	Configure a RD for VRF; RD information will be need when BGP distributes VPN routes and MDT-SAFI routes.
Ruijie(config-vrf)# **route-target** {**both | export | import}** *rt-value*	Configure RT value.
Ruijie(config-vrf)# **mdt default** *group_address*	Configure Default-MDT: specify the group address and enable MD.

Note

After executing "**mdt default**" command, a MTI interface will be generated automatically. This MTI interface can only be dynamically generated by MD control plane instead of user configuration.

Note

Execute "**neighbor** *ip-address* **activate**" command under vpnv4 address family; after establishing the connection between device and neighbor, MTI interface will take the interface automatically acquired to establish connection with neighbor as the source interface, and take the source address to establish connection with neighbor as its IP address.

Note

After configuration, the group address of Default-MDT cannot be changed by running the command. To change the group address, you can only delete the existing one and configure a new one. If you run the **no mdt default** command to delete Default-MDT, the MTI interface and the configuration command of Data-MDT corresponding to the VRF will also be deleted.

# Configure Default-MDT

```
Ruijie# configure terminal
Ruijie(config)# ip vrf vpn1
Ruijie(config-vrf)# rd 1: 100
Ruijie(config-vrf)# route-target both 1: 100
Ruijie(config-vrf)# mdt default 239.1.1.1
```

Configuring Data-MDT

To configure Data-MDT forwarding for some high-rate private network multicast traffic, you need to configure Data-MDT under VRF mode, as shown below:

Command	Function
Ruijie# **configure terminal**	Enter global configuration mode.
Ruijie(config)# **ip vrf** *vrf_name*	Create a VRF and enter VRF configuration mode.

Ruijie(config-vrf)# **mdt data** group_address wildcard_bits [**list** access-list]	Configure Data-MDT.

Note

After executing "mdt data" command, all private network multicast traffic meeting ACL filtering conditions will reach PE, which will create a Data-MDT. Each PE can only create a limited number of Data-MDT. Therefore, it is suggested that Data-MDT can only be configured for certain high-rate private network multicast traffic.

Note

Data-MDT cannot be reused, i.e., one Data-MDT can only be used to forward one private network multicast traffic.

# Configure Data-MDT, so that the private network multicast traffic specified by ACL will be forwarded with Data-MDT.

```
Ruijie(config)# ip vrf vpn1
Ruijie(config-vrf)# mdt data 238.1.1.0 0.0.0.255 list vpn1-datamdt
Ruijie(config-vrf)# exit
Ruijie(config-vrf)# ip access-list standard vpn1-datamdt
Ruijie(config-std-nacl)# permit 224.1.1.0 0.0.0.255
```

Configuring Multicast Routing on VRF

Configuration of multicast routing on VRF will need to enable multicast routing on VRF and configure the PIM-SM instance on this VRF. When vrf parameter is not specified, the following commands shall mean the configuration of public network multicast routing. The following table describes the procedure.

Command	Function
Ruijie# **configure terminal**	Enter global configuration mode.
Ruijie(config)# **ip multicast-routing vrf** *vrf_name*	Enable multicast routing of certain VRF.
Ruijie(config)# **ip pim** [**vrf** *vrf-name*] **rp-address** *rp-address* [*access_list*]	Configure static RP on certain VRF.
Ruijie(config)**#** **interface** *type ID*	Configure interface.
Ruijie(config-if-*type ID*)# **ip vrf forwarding** *vrf_name*	Bind the interface with certain VRF.
Ruijie(config-if-*type ID*)# **ip address** *ip-address mask*	Configure IP address on interface.
Ruijie(config-if-*type ID*)# **ip pim sparse-mode**	Enable PIM-SM on interface.

# Configure multicast routing

```
Ruijie# configure terminal
Ruijie(config)# ip multicast-routing
Ruijie(config)# ip pim rp-address 10.1.1.2
Ruijie(config)# interface loopback 0
Ruijie(config-Loopback 0)# ip address 10.1.1.1 255.255.255.255
Ruijie(config-Loopback 0)# ip pim sparse-mode
Ruijie(config-Loopback 0)# exit
Ruijie(config)# ip multicast-routing vrf vpn1
Ruijie(config)# ip pim vrf vpn1 rp-address 1.1.1.1
Ruijie(config)# interface gigabitethernet 0/1
```

# "no switchport" command is used to switch the port mode to "Routed Port" mode on switch products, and is not applicable to the router. Therefore, you don't need to execute this command on router products.

```
Ruijie(config-GigabitEthernet 0/1)# no switchport
Ruijie(config-GigabitEthernet 0/1)# ip vrf forwarding vpn1
Ruijie(config-GigabitEthernet 0/1)# ip address 1.1.1.1 255.255.255.0
Ruijie(config-GigabitEthernet 0/1)# ip pim sparse-mode
```

Configuring PIM-SSM

This step intends to create Default-MDT in order to use PIM-SSM, as shown below:

Command	Function
Ruijie# **configure terminal**	Enter global configuration mode.
Ruijie(config)# **ip pim ssm** { **default** | **range** *access_list*}	Configure the scope of PIM-SSM. After configuring PIM-SSM, the group address in "mdt default" command must fall within the scope of SSM. Only in this way we can use PIM-SSM to create Default-MDT.

# Configure the scope of SSM

```
Ruijie# configure terminal
Ruijie(config)# ip pim ssm default
```

Configuring BGP MDT Address Family

When using PIM-SSM to create Default-MDT, you will need to configure BGP MDT address family. Through the routing of MDT address family, PE can discover other PE addresses and initiate the grating of SPT to other PEs. The configuration steps of MDT address family are shown below:

Command	Function
Ruijie# **configure terminal**	Enter global configuration mode.
Ruijie(config)# **router bgp** *asn-num*	Create BGP and enter BGP configuration mode.
Ruijie(config-router)# **neighbor** *ip-address* **remote-as** *asn-number*	Configure BGP session.
Ruijie(config-router)# **neighbor** *ip-address* **update-source** *interface-name*	Configure to use interface address as the source address when MP-IBGP session is established; usually, the Loopback interface address is used as the source address.
Ruijie(config-router)# **address-family ipv4 mdt**	Enter MDT address family.
Ruijie(config-router-af)# **neighbor** *ip-address* **activate**	Activate the route to exchange MDT address family on BGP session.
Ruijie(config-router-af)# **neighbor** *ip-address* **next-hop-self**	Change the next-hop route to self; this command can be executed on ASBR in OptionB.

# Configure MDT address family

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 10.1.1.3 remote-as 2
Ruijie(config-router)# neighbor 10.1.1.3 update-source loopback 0
Ruijie(config-router)# address-family ipv4 mdt
Ruijie(config-router-af)# neighobr 10.1.1.3 activate
```

Configuring Multi-AS Multicast VPN

The configuration of multi-AS multicast VPN could be different according to different unicast schemes. In OptionA, the configuration of multicast VPN is similar to that of single AS, and is hence not introduced herein. We will mainly introduce the configurations of OptionB and OptionC. Before configuring multi-AS multicast VPN, the following network configurations must have been completed:

- Configure multi-AS BGP/MPLS L3VPN
- Configure Default-MDT

The configuration of multi-AS multicast VPN mainly involves the following:

- Multicast VPN configuration when using PIM-SM
- Multicast VPN configuration in OptionB when using PIM-SSM
- Multicast VPN configuration in OptionC when using PIM-SSM

Multicast VPN Configuration when Using PIM-SM

When configuring MSDP-based multi-AS multicast VPN, the multicast configuration on PE shall be the same as the configuration to create Default-MDT with PIM-SM. MSDP must be configured on ASBR, as shown below:

Command	Function
Ruijie# **configure terminal**	Enter global configuration mode.
Ruijie(config)# **ip msdp peer** *peer-address* **connect-source** *interface-type interface-number*	Configure MSDP peer.

# Configure MSDP peer

```
Ruijie# configure terminal
Ruijie(config)# ip msdp peer 10.0.5.9 connect-source gi 0/1
```

Multicast VPN Configuration in OptionB when Using PIM-SSM

Configure multicast VPN in OptionB, so that it can use PIM-SSM to create Default-MDT, as shown below:

Command	Function
Ruijie# **configure terminal**	Enter global configuration mode.
Ruijie(config)# **router bgp** *asn-num*	Create BGP and enter BGP configuration mode.
Ruijie(config-router)# **neighbor** *ip-address* **remote-as** *asn-number*	Configure BGP session.
Ruijie(config-router)# **neighbor** *ip-address* **update-source** *interface-name*	Configure to use interface address as the source address when BGP session is established; usually, the Loopback interface address is used as the source address.
Ruijie(config-router)# **address-family ipv4 mdt**	Enter MDT address family.

Ruijie(config-router-af)# **neighbor** *ip-address* **activate**	Activate the route to exchange MDT-SAFI on BGP session.
Ruijie(config-router-af)# **exit**	Exit the address family.
Ruijie(config-router)# **exit**	Exit BGP mode.
Ruijie(config)# **ip multicast** [**vrf** *vrf-name*] **rpf proxy rd vector**	Configure whether to enable RPF Vector on certain VRF. In OptionB, the rd parameter of this command must be configured to carry RD information in Join attribute. This is because in OptionB, the next-hop of route will change when MDT-SAFI route passes the ASBR. Therefore, the next hop may be different for route entries with same destination address but different group addresses. By this time, we will need to use RD+ destination address as the index for discovering routes.
Ruijie(config)**# ip pim ssm default**	Configure PIM-SSM.

# Configure multicast VPN in OptionB

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 10.1.1.3 remote-as 1
Ruijie(config-router)# neighbor 10.1.1.3 update-source loopback 0
Ruijie(config-router)# address-family ipv4 mdt
Ruijie(config-router-af)# neighbor 10.1.1.3 activate
Ruijie(config-router-af)# neighbor 10.1.1.3 next-hop-self
Ruijie(config-router-af)# exit
Ruijie(config-router)# exit
Ruijie(config)# ip multicast vrf vpn1 rpf proxy rd vector
Ruijie(config)# ip pim ssm default
```

Multicast VPN Configuration in OptionC when Using PIM-SSM

Configure multicast VPN in OptionC, so that it can use PIM-SSM to create Default-MDT, as shown below:

Command	Function
Ruijie# **configure terminal**	Enter global configuration mode.
Ruijie (config)# **router bgp** *asn-num*	Create BGP and enter BGP configuration mode.
Ruijie (config-router)# **neighbor** *ip-address* **remote-as** *asn-number*	Configure BGP session.
Ruijie(config-router)# **neighbor** *ip-address* **update-source** *interface-name*	Configure to use interface address as the source address when BGP session is established; usually, the Loopback interface address is used as the source address.

Ruijie(config-router)# **neighbor** *ebgp-peer-address* **ebgp-multihop**	Configure multi-hop attribute.
Ruijie (config-router)# **address-family ipv4 mdt**	Enter MDT address family.
Ruijie (config-router-af)# **neighbor** *ip-address* **activate**	Activate the route to exchange MDT-SAFI on BGP session.
Ruijie (config-router-af)# **exit**	Exit the address family.
Ruijie (config-router)# **exit**	Exit BGP mode.
Ruijie (config)# **ip multicast rpf proxy vector**	Enable RPF Vector.
Ruijie (config)# **ip pim ssm default**	Configure PIM-SSM.

# Configure multicast VPN in OptionC.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 10.1.1.5 remote-as 2
Ruijie(config-router)# neighbor 10.1.1.5 update-source loopback 0
Ruijie(config-router)# address-family ipv4 mdt
Ruijie(config-router-af)# neighbor 10.1.1.5 activate
Ruijie(config-router-af)# exit
Ruijie(config-router)# exit
Ruijie(config)# ip multicast rpf proxy vector
Ruijie(config)# ip pim ssm default
```

Extranet-MVPNConfiguring Extranet-MVPN

Ensure that the following network configurations are completed before configuring Extranet-MPVN:

- Configuration of single-AS or cross-AS multicast VPN

To ensure that the multicast stream can be transmitted from the ingress PE to the egress PE on the public network, the following two ways to configure Extranet-MVPN:

- Configuring the receiver MVRF on the ingress PE.
- Configuring the source MVRF on the egress PE.

Configuring the Receiver MVRF on the Ingress PE

After this method is used, the ingress PE is added to the Mobile Data Terminal (MDT) of the receiver MVRF. In this case, the ingress PE can send the multicast stream to the public network MDT of the egress PE, and eventually to the egress PE. The following table describes the procedure.

Command	Function
Ruijie# **configure terminal**	Enter the global configuration mode.
Ruijie(config)# **ip vrf** *vrf_name*	Create a VRF and enter the VRF configuration mode.
Ruijie(config-vrf)# **rd** *route_distinguisher*	Configure an RD, BGP distribution VPN routing, and RD information used during MDT-SAFI routing for the VRF.
Ruijie(config-vrf)# **route-target import** *rt-value*	Configure the RT value to be imported. Ensure that the receiver MVRF of the ingress PE imports the VPN route where the multicast source resides, so that the PE can create the cross-VRF multicast distribution tree.
Ruijie(config-vrf)# **mdt default** *group_address*	Configure Default-MDT. The command specifies the group address and enables the MD. The group address must be the same as that of the MVRF of the egress PE.

If multiple receiver VPNs exist, you need to configure multiple receiver MVRFs on the ingress PE. When forwarding the multicast stream, the ingress PE makes a copy of the multicast steam and broadcasts it on the public network for the MDT of each receiver MVRF. If the traffic of the multicast steam is high, making copies of the multicast stream will cause heave waste of the public network's bandwidth. In this case, you can configure the source MVRF on the egress PE. The method saves the trouble of configuring the source MVRF on all egress PEs. However, it may waste bandwidth. Therefore, it should only be used when the traffic of the multicast stream is low.

# Configure the receiver MVRF.

```
Ruijie# configure terminal
Ruijie(config)# ip vrf vpn2
Ruijie(config-vrf)# rd 2: 200
Ruijie(config-vrf)# route-target import 1: 100
Ruijie(config-vrf)# mdt default 239.2.2.2
```

Configuring the Source MVRF on the Egress PE

This method is to configure the MVRF consistent with the ingress PE on the egress PE, so that the egress PE can join the public-network MDT of the source MVRF and receive the multicast stream sent by the ingress PE on the public network.

The following table describes the procedure.

Command	Function
Ruijie# **configure terminal**	Enter the global configuration mode.
Ruijie(config)# **ip vrf** *vrf_name*	Create a VRF and enter the VRF configuration mode.
Ruijie(config-vrf)# **rd** *route_distinguisher*	Configure an RD, BGP distribution VPN routing, and RD information used during MDT-SAFI routing for the VRF.

Ruijie(config-vrf)# **mdt default** *group_address*	Configure Default-MDT. This command specifies the group address and enables the MD. The group address must be the same as that of the MVRF of the ingress PE.
Ruijie(config-vrf)# **ip mroute** [**vrf** *vrf-name*] *source-address mask* **fallback-lookup** { **global** | **vrf** *vrf-name* } [*distance*]	Configure a static multicast route and specify the VRF to be selected during the RPF check. Ensure that the VRF is consistent with the source MVRF.

This method is recommended for saving public-network bandwidth when the traffic of the multicast stream send by the multicast source is high. Pay attention to the following points when using this method:

- You must configure the source MVRF on all egress PEs. The configuration is more complicated than configuring the receiver PE on the ingress PE.
- When querying the multicast RPF on the VRF (connected to the receiver VPN) of the egress PE, you need to locate the source MVRF and use the MDT of the source MVRF to send the multicast receiving request to the ingress PE through the public network.

# Configure the source MVRF.

```
Ruijie# configure terminal
Ruijie(config)# ip vrf vpn1
Ruijie(config-vrf)# rd 1: 100
Ruijie(config-vrf)# route-target import 1: 100
Ruijie(config-vrf)# mdt default 239.1.1.1
Ruijie(config-vrf)# ip mroute vrf vpn2 192.168.0.0 255.255.255.0 fallback-lookup vrf vpn1
```

Configuring the VRF to Be Selected by the Static Multicast Route

The unicast route is used in the RPF check by default. Therefore, the same VRF is used in the multicast RPF check and unicast routing. You can configure the static multicast route to change default RPF configuration for distinguishing the topology of the multicast route from the topology of the unicast route and enabling the unicast and multicast routes to use different VRFs.

Specifically, you need to specify a VRF (not used by the unicast route) to be used by the static multicast route for the multicast RPF check on the network segment. The VRF must be different from that used by the unicast route.

This configuration method allows the multicast stream of a VPN site to be released on the public network. BGP cannot deliver a VPN route to the routing table on the public network. As a result, the RPF check on the public network cannot locate the next-hop leading to the multicast source of the VPN site. The configuration of the static multicast route, however, allows specifying a VRF used in the RPF check. Thereafter, the next-hop leading to the multicast source of the VPN site can be located.

The following table describes the procedure.

Command	Function
Ruijie# **configure terminal**	Enter the global configuration mode.

Ruijie(config)# **ip mroute** [**vrf** *vrf-name*] *source-address mask* **fallback-lookup** {**global | vrf** *source-vrf-name*} [*distance*]	Specify a VRF in the static multicast route. Set vrf *vrf-name* to the VRF where this command functions for the RPF check. Set *source-address mask* to the route prefix and mask for specifying the route range where the command functions for the RPF check. Use the keyword global to indicate that the RPF check is performed on the public network. Set vrf *source-vrf-name* to the VRF used in the RPF check. Set *distance* to the distance value of the route.
Ruijie(config)# **end**	Exit the configuration mode.
Ruijie# **show ip mroute** [**vrf** *vrf-name*] static	Show the configuration result of static multicast route.

# Specify the VRF to be selected by the static multicast route.

```
Ruijie# configure terminal
Ruijie(config)# ip mroute vrf vpn1 1.1.1.0 255.255.255.0 fallback-lookup vrf vpn2
Ruijie(config)# end
```

Verifying the Operating Status of Multicast VPN

This section will introduce how to verify the configurations of multicast VPN and routing information. Enter privilege mode and execute the following commands:

Command	Function
Ruijie# **show ip pim mdt** [**bgp**]	Display MDT information.
Ruijie# **show ip pim vrf** *vrf-name* **mdt send**	Display the sending status of MDT-Join message.
Ruijie# **show ip pim vrf** *vrf-name* **mdt receive**	Display the reception status of MDT-Join message.
Ruijie# **show ip pim sparse-mode** [**vrf** *vrf-name*] **neighbor [detail]**	Display the establishment status of PIM-SM neighbor.
Ruijie# **show ip pim sparse-mode mroute proxy**	Display the RPF Vector information of PIM-SM.

# Display MDT information

```
Ruijie# show ip pim mdt
* implies group is the MDT default group
  MDT Group      Interface   Source                 VRF
* 239.1.1.1      Tunnel0     Loopback0               vpn1
```

# Display the sending and reception information of MDT-Join packets.

```
Ruijie# show ip pim vrf vpn1 mdt send
MDT-data send list for VRF: vpn1
  (source, group)                   MDT-data group
  (192.168.1.107, 224.1.1.1)          239.1.2.0

Ruijie# show ip pim vrf vpn1 mdt receive
```

```
Joined MDT-data [group : source]  uptime/expires for VRF: vpn1
 [239.1.2.0 : 0.0.0.0]  00:00:28/00:02:35
```

# Display PIM-SM neighbor

```
Ruijie# show ip pim sparse-mode vrf vpn1 neighbor
Neighbor    Interface          Uptime/Expires     Ver   DR
Address                                                Priority/Mode
1.1.1.20    FastEthernet 0/1   00:02:25/00:01:20   v2   1 / P
```

# Display PIM-SM RPF Vector

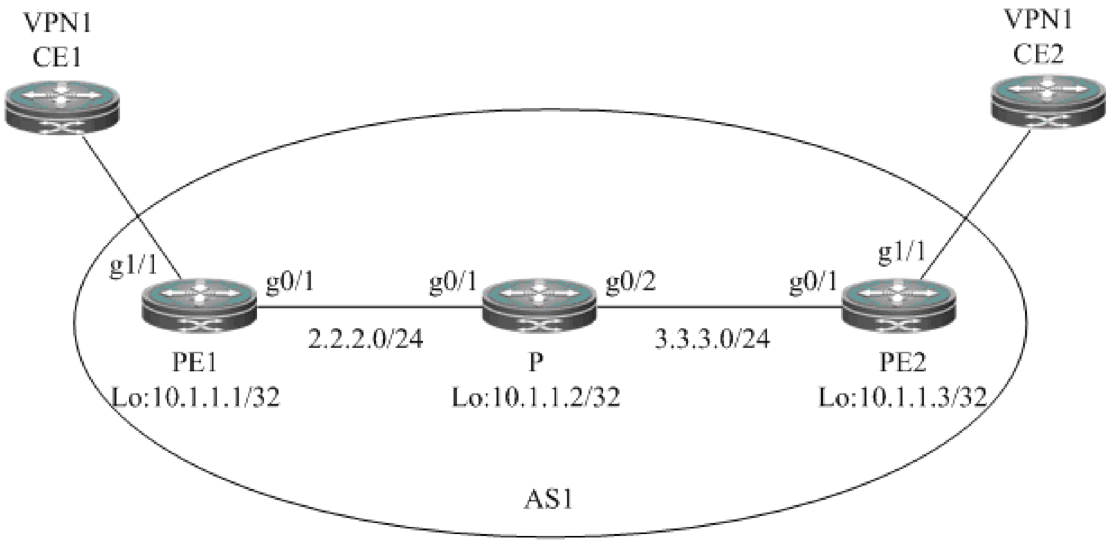
```
Ruijie# show ip pim sparse-mode mroute proxy
(20.1.1.1, 232.1.1.1)
  Proxy                    Assigner        Origin   Uptime/Expire
  1:100/10.1.1.3           0.0.0.0         BGP MDT   00:09:25/stopped
```

Multicast VPN Configuration Example

Example of Single-AS Multicast VPN Configuration

Requirement: The unicast routing has been configured for VPN and public network, and MPLS has also been configured for the public network, with topology shown below:

Figure 27 Topology of single-AS multicast VPN



Configuration steps:

- CE1:

# Configure multicast routing

```
Ruijie# configure terminal
Ruijie(config)# ip multicast-routing
Ruijie(config)# ip pim rp-address 192.168.0.1
Ruijie(config)# interface gigabitethernet 0/1
Ruijie(config-GigabitEthernet 0/1)# ip address 192.168.0.1 255.255.255.0
Ruijie(config-GigabitEthernet 0/1)# ip pim sparse-mode
```

- PE1:

# Configure VRF

Create a VRF of "vpn1", define RD value and RT value and configure Default-MDT.

```
Ruijie# configure terminal
Ruijie(config)# ip vrf vpn1
Ruijie(config-vrf)# rd 1:100
Ruijie(config-vrf)# route-target both 1:100
Ruijie(config-vrf)# mdt default 239.1.1.1
Ruijie(config-vrf)# end
```

# Configure private network multicast routing

```
Ruijie# configure terminal
Ruijie(config)# ip multicast-routing vrf vpn1
Ruijie(config)# ip pim vrf vpn1 rp-address 192.168.0.1
Ruijie(config)# interface gigabitethernet 1/1
```

# "no switchport" command is used to switch the port mode to "Routed Port" mode on switch products, and is not applicable to the router. Therefore, you don't need to execute this command on router products.

```
Ruijie(config-GigabitEthernet 1/1)# no switchport
Ruijie(config-GigabitEthernet 1/1)# ip vrf forwarding vpn1
Ruijie(config-GigabitEthernet 1/1)# ip address 192.168.0.2 255.255.255.0
Ruijie(config-GigabitEthernet 1/1)# ip pim sparse-mode
Ruijie(config-GigabitEthernet 1/1)# end
```

# Configure public network multicast routing

```
Ruijie# configure terminal
Ruijie(config)# ip multicast-routing
Ruijie(config)# ip pim rp-address 10.1.1.2
Ruijie(config)# interface gigabitethernet 0/1
```

# "no switchport" command is used to switch the port mode to "Routed Port" mode on switch products, and is not applicable to the router. Therefore, you don't need to execute this command on router products.

```
Ruijie(config-GigabitEthernet 0/1)# no switchport
Ruijie(config-GigabitEthernet 0/1)# ip address 2.2.2.1 255.255.255.0
Ruijie(config-GigabitEthernet 0/1)# ip pim sparse-mode
Ruijie(config-GigabitEthernet 0/1)# exit
Ruijie(config)# interface loopback 0
Ruijie(config-Loopback 0)# ip address 10.1.1.1 255.255.255.255
Ruijie(config-Loopback 0)# ip pim sparse-mode
Ruijie(config-Loopback 0)# end
```

# Configure BGP protocol to establish IBGP session with PE2 and configure VPNv4 address family

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 10.1.1.3 remote-as 1
Ruijie(config-router)# neighbor 10.1.1.3 update-source loopback 0
Ruijie(config-router)# address-family vpnv4
Ruijie(config-router-af)# neighbor 10.1.1.3 activate
Ruijie(config-router-af)# neighbor 10.1.1.3 send-community extended
Ruijie(config-router-af)# end
```

- P:

# Configure public network multicast routing

```
Ruijie# configure terminal
Ruijie(config)# ip multicast-routing
Ruijie(config)# ip pim rp-address 10.1.1.2
Ruijie(config)# interface gigabitethernet 0/1
```

# "no switchport" command is used to switch the port mode to "Routed Port" mode on switch products, and is not applicable to the router. Therefore, you don't need to execute this command on router products.

```
Ruijie(config-GigabitEthernet 0/1)# no switchport
Ruijie(config-GigabitEthernet 0/1)# ip address 2.2.2.2 255.255.255.0
Ruijie(config-GigabitEthernet 0/1)# ip pim sparse-mode
Ruijie(config-GigabitEthernet 0/1)# exit
Ruijie(config)# interface gigabitethernet 0/2
```

# "no switchport" command is used to switch the port mode to "Routed Port" mode on switch products, and is not applicable to the router. Therefore, you don't need to execute this command on router products.

```
Ruijie(config-GigabitEthernet 0/2)# no switchport
Ruijie(config-GigabitEthernet 0/2)# ip address 3.3.3.1 255.255.255.0
Ruijie(config-GigabitEthernet 0/2)# ip pim sparse-mode
Ruijie(config-GigabitEthernet 0/2)# exit
Ruijie(config)# interface loopback 0
Ruijie(config-Loopback 0)# ip address 10.1.1.2 255.255.255.255
Ruijie(config-Loopback 0)# ip pim sparse-mode
```

- PE3:

The configuration steps are similar to that of PE1.

- CE2:

The configuration steps are similar to that of CE1.

- Verify configurations:

# Execute "**show ip pim mdt**" command on PE1 to display MDT established.

```
Ruijie# show ip pim mdt
```

```
* implies group is the MDT default group
  MDT Group       Interface   Source                VRF
* 239.1.1.1       Tunnel0     Loopback0             vpn1
```

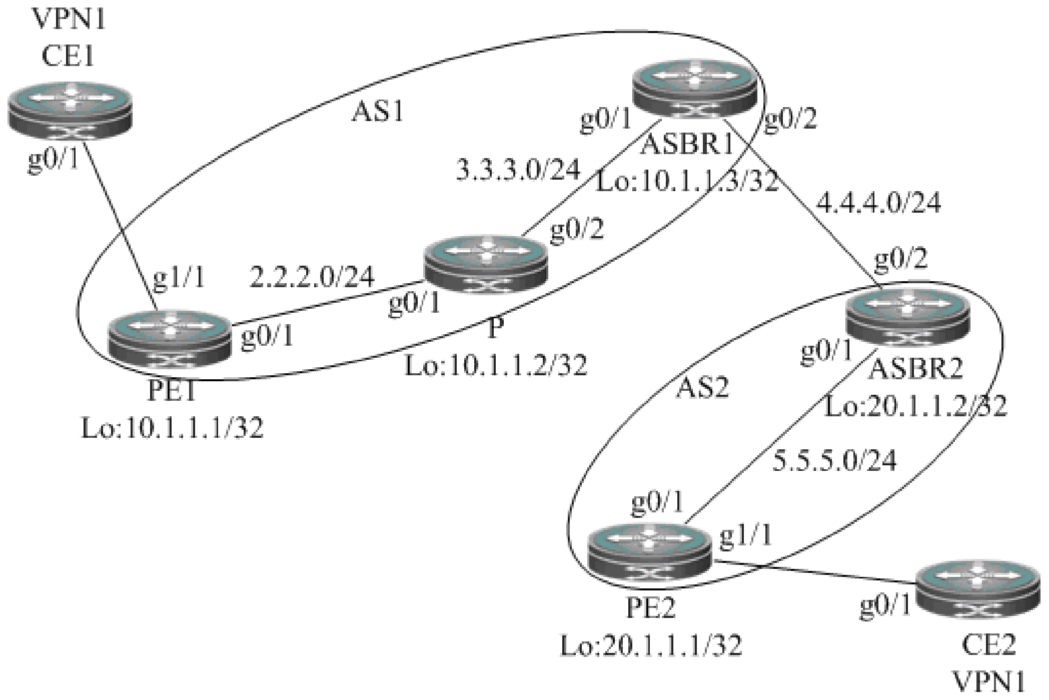
# Execute "**show ip pim sparse-mode neighbor**" command on PE1 to display the neighbors established under VRF.

```
Ruijie# show ip pim sparse-mode vrf vpn1 neighbor
Neighbor    Interface           Uptime/Expires     Ver   DR
Address                                                 Priority/Mode
192.168.0.1 Gigabitethernet 0/1 00:00:01/00:01:44  v2    1 / P
10.1.1.3     Tunnel0             00:02:25/00:01:20  v2    1 / P
```

Example of Multicast VPN Configuration in OptionC when Using PIM-SM

Requirement: BGP/MPLS L3VPN has been configured according to the scheme of OptionC, and PE route has been distributed to respective ASes through BGP.

Figure 28 Topology of multi-AS multicast VPN



Configuration steps:

- CE1:

The configuration steps are similar to that of CE1 in the scheme of single-AS multicast VPN.

- CE2:

The configuration steps are similar to that of CE2 in the scheme of single-AS multicast VPN.

- PE1:

# Configure VRF

Create a VRF of "vpn1", define RD value and RT value and configure Default-MDT.

```
Ruijie# configure terminal
Ruijie(config)# ip vrf vpn1
Ruijie(config-vrf)# rd 1:100
Ruijie(config-vrf)# route-target both 1:100
Ruijie(config-vrf)# mdt default 239.1.1.1
Ruijie(config-vrf)# end
```

# Configure private network multicast routing

```
Ruijie# configure terminal
Ruijie(config)# ip multicast-routing vrf vpn1
Ruijie(config)# ip pim vrf vpn1 rp-address 192.168.0.1
Ruijie(config)# interface gigabitethernet 1/1
```

# "no switchport" command is used to switch the port mode to "Routed Port" mode on switch products, and is not applicable to the router. Therefore, you don't need to execute this command on router products.

```
Ruijie(config-GigabitEthernet 1/1)# no switchport
Ruijie(config-GigabitEthernet 1/1)# ip vrf forwarding vpn1
Ruijie(config-GigabitEthernet 1/1)# ip address 192.168.0.2 255.255.255.0
Ruijie(config-GigabitEthernet 1/1)# ip pim sparse-mode
Ruijie(config-GigabitEthernet 1/1)# end
```

# Configure public network multicast routing

```
Ruijie# configure terminal
Ruijie(config)# ip multicast-routing
Ruijie(config)# interface gigabitethernet 0/1
```

# "no switchport" command is used to switch the port mode to "Routed Port" mode on switch products, and is not applicable to the router. Therefore, you don't need to execute this command on router products.

```
Ruijie(config-GigabitEthernet 0/1)# no switchport
Ruijie(config-GigabitEthernet 0/1)# ip address 2.2.2.1 255.255.255.0
Ruijie(config-GigabitEthernet 0/1)# ip pim sparse-mode
Ruijie(config-GigabitEthernet 0/1)# exit
Ruijie(config)# interface loopback 0
Ruijie(config-Loopback 0)# ip address 10.1.1.1 255.255.255.255
Ruijie(config-Loopback 0)# ip pim sparse-mode
Ruijie(config-Loopback 0)# end
```

- P:

The configuration steps are similar to that of P in the scheme of single-AS multicast VPN.

- ASBR1:

# Configure public network multicast routing

```
Ruijie# configure terminal
Ruijie(config)# ip multicast-routing
Ruijie(config)# interface gigabitethernet 0/1
```

# "no switchport" command is used to switch the port mode to "Routed Port" mode on switch products, and is not applicable to the router. Therefore, you don't need to execute this command on router products.

```
Ruijie(config-GigabitEthernet 0/1)# no switchport
Ruijie(config-GigabitEthernet 0/1)# ip address 3.3.3.2 255.255.255.0
Ruijie(config-GigabitEthernet 0/1)# ip pim sparse-mode
Ruijie(config-GigabitEthernet 0/1)# exit
Ruijie(config)# interface gigabitethernet 0/2
```

# "no switchport" command is used to switch the port mode to "Routed Port" mode on switch products, and is not applicable to the router. Therefore, you don't need to execute this command on router products.

```
Ruijie(config-GigabitEthernet 0/2)# no switchport
Ruijie(config-GigabitEthernet 0/2)# ip address 4.4.4.1 255.255.255.0
Ruijie(config-GigabitEthernet 0/2)# ip pim sparse-mode
Ruijie(config-GigabitEthernet 0/2)# exit
Ruijie(config)# interface loopback 0
Ruijie(config-Loopback 0)# ip address 10.1.1.3 255.255.255.255
Ruijie(config-Loopback 0)# ip pim sparse-mode
Ruijie(config-Loopback 0)# end
```

# Configure candidate BSR, candidate RP and MSDP Peer, and configure BSR boundary on the interface connecting with ASBR2, so that two unicast domains can be isolated to different RP domains.

```
Ruijie# configure terminal
Ruijie(config)# ip pim bsr-candidate loopback 0
Ruijie(config)# ip pim rp-candidate loopback 0
Ruijie(config)# ip msdp peer 4.4.4.2 connect-source gi 0/2
Ruijie(config)# interface gigabitethernet 0/2
Ruijie(config-GigabitEthernet 0/2)# ip pim bsr-border
```

- ASBR2:

The configuration steps are similar to that of ASBR1.

- PE2:

The configuration steps are similar to that of PE1.

- Verify configurations:

# Execute "**show ip pim mdt**" command on PE1 to display the MDT established.

```
Ruijie# show ip pim mdt
* implies group is the MDT default group
  MDT Group      Interface   Source                VRF
* 239.1.1.1      Tunnel0     Loopback0             vpn1
```

# Execute "**show ip pim sparse-mode neighbor**" command on PE1 to display the neighbors established under VRF.

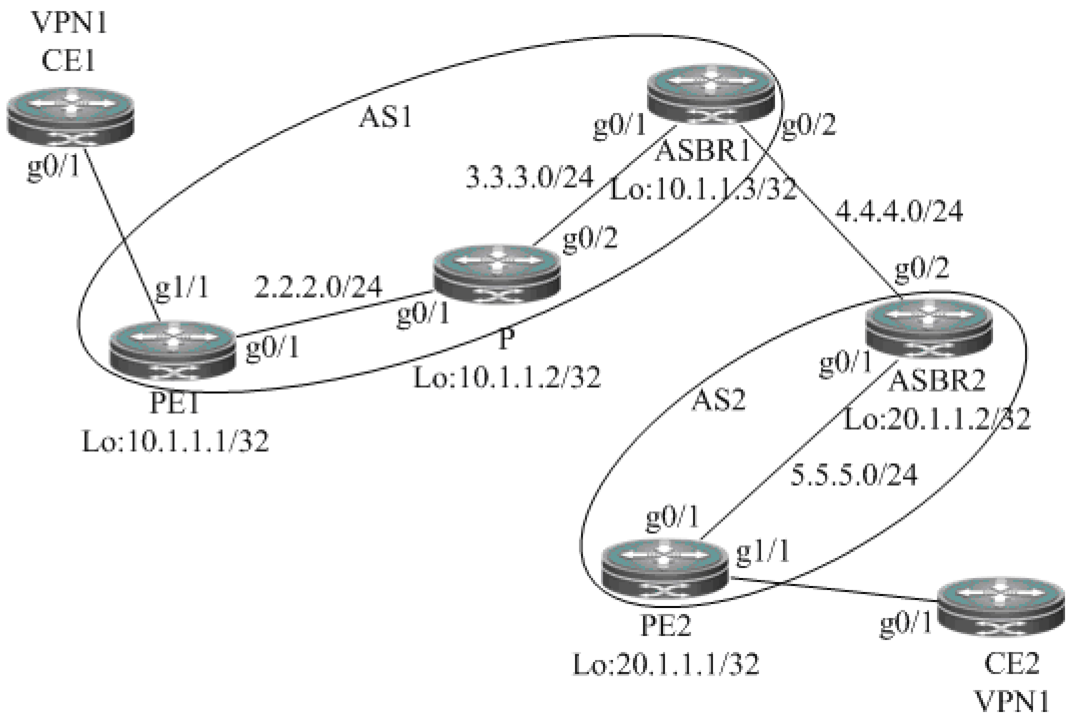
```
Ruijie# show ip pim sparse-mode vrf vpn1 neighbor
```

```
Neighbor    Interface           Uptime/Expires     Ver   DR
Address                                                   Priority/Mode
192.168.0.1 Gigabitethernet 0/1 02:34:20/00:01:31  v2    1 / P
20.1.1.1     Tunnel0             01:55:19/00:01:37  v2    1 / P
```

Example of Multicast VPN Configuration in OptionB when Using PIM-SSM

Requirement: BGP/MPLS L3VPN has been configured according to the scheme of OptionB.

Figure 29 Topology of multi-AS multicast VPN (OptionB)



Configuration steps:

- CE1:

The configuration steps are similar to that of CE1 in the scheme of single-AS multicast VPN.

- CE2:

The configuration steps are similar to that of CE2 in the scheme of single-AS multicast VPN.

- PE1:

# Configure VRF

Create a VRF of "vpn1", define RD value and RT value and configure Default-MDT. The group address of Default-MDT must fall within the scope of PIM-SSM.

```
Ruijie# configure terminal
Ruijie(config)# ip vrf vpn1
Ruijie(config-vrf)# rd 1:100
Ruijie(config-vrf)# route-target both 1:100
Ruijie(config-vrf)# mdt default 232.1.1.1
Ruijie(config-vrf)# end
```

# Configure private network multicast routing

```
Ruijie# configure terminal
Ruijie(config)# ip multicast-routing vrf vpn1
Ruijie(config)# ip pim vrf vpn1 rp-address 192.168.0.1
Ruijie(config)# interface gigabitethernet 1/1
```

# "no switchport" command is used to switch the port mode to "Routed Port" mode on switch products, and is not applicable to the router. Therefore, you don't need to execute this command on router products.

```
Ruijie(config-GigabitEthernet 1/1)# no switchport
Ruijie(config-GigabitEthernet 1/1)# ip vrf forwarding vpn1
Ruijie(config-GigabitEthernet 1/1)# ip address 192.168.0.2 255.255.255.0
Ruijie(config-GigabitEthernet 1/1)# ip pim sparse-mode
Ruijie(config-GigabitEthernet 1/1)# end
```

# Configure public network multicast routing, and configure to use RPF Vector that carries RD information.

```
Ruijie# configure terminal
Ruijie(config)# ip multicast-routing
Ruijie(config)# ip multicast vrf vpn1 rpf proxy rd vector
Ruijie(config)# ip pim ssm default
Ruijie(config)# interface gigabitethernet 0/1
```

# "no switchport" command is used to switch the port mode to "Routed Port" mode on switch products, and is not applicable to the router. Therefore, you don't need to execute this command on router products.

```
Ruijie(config-GigabitEthernet 0/1)# no switchport
Ruijie(config-GigabitEthernet 0/1)# ip address 2.2.2.1 255.255.255.0
Ruijie(config-GigabitEthernet 0/1)# ip pim sparse-mode
Ruijie(config-GigabitEthernet 0/1)# exit
Ruijie(config)# interface loopback 0
Ruijie(config-Loopback 0)# ip address 10.1.1.1 255.255.255.255
Ruijie(config-Loopback 0)# ip pim sparse-mode
Ruijie(config-Loopback 0)# end
```

# Configure BGP protocol to establish IBGP session with ASBR1 and configure MDT address family

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 10.1.1.3 remote-as 1
Ruijie(config-router)# neighbor 10.1.1.3 update-source loopback 0
Ruijie(config-router)# address-family ipv4 mdt
Ruijie(config-router-af)# neighbor 10.1.1.3 activate
Ruijie(config-router-af)# end
```

- P:

The configuration steps are similar to that of P in the scheme of single-AS multicast VPN.

- ASBR1:

# Configure public network multicast routing

```
Ruijie# configure terminal
Ruijie(config)# ip multicast-routing
Ruijie(config)# interface gigabitethernet 0/1
```

# "no switchport" command is used to switch the port mode to "Routed Port" mode on switch products, and is not applicable to the router. Therefore, you don't need to execute this command on router products.

```
Ruijie(config-GigabitEthernet 0/1)# no switchport
Ruijie(config-GigabitEthernet 0/1)# ip address 3.3.3.2 255.255.255.0
Ruijie(config-GigabitEthernet 0/1)# ip pim sparse-mode
Ruijie(config-GigabitEthernet 0/1)# exit
Ruijie(config)# interface gigabitethernet 0/2
```

# "no switchport" command is used to switch the port mode to "Routed Port" mode on switch products, and is not applicable to the router. Therefore, you don't need to execute this command on router products.

```
Ruijie(config-GigabitEthernet 0/2)# no switchport
Ruijie(config-GigabitEthernet 0/2)# ip address 4.4.4.1 255.255.255.0
Ruijie(config-GigabitEthernet 0/2)# ip pim sparse-mode
Ruijie(config-GigabitEthernet 0/2)# exit
Ruijie(config)# interface loopback 0
Ruijie(config-Loopback 0)# ip address 10.1.1.3 255.255.255.255
Ruijie(config-Loopback 0)# ip pim sparse-mode
Ruijie(config-Loopback 0)# end
```

# Configure BGP protocol to establish IBGP session with PE1 and EBGP session with ASBR2, and configure MDT address family

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 10.1.1.1 remote-as 1
Ruijie(config-router)# neighbor 10.1.1.1 update-source loopback 0
Ruijie(config-router)# neighbor 4.4.4.2 remote-as 2
Ruijie(config-router)# address-family ipv4 mdt
Ruijie(config-router-af)# neighbor 10.1.1.1 activate
Ruijie(config-router-af)# neighbor 4.4.4.2 activate
```

- ASBR2:

# Configure public network multicast routing

```
Ruijie# configure terminal
Ruijie(config)# ip multicast-routing
Ruijie(config)# interface gigabitethernet 0/1
```

# "no switchport" command is used to switch the port mode to "Routed Port" mode on switch products, and is not applicable to the router. Therefore, you don't need to execute this command on router products.

```
Ruijie(config-GigabitEthernet 0/1)# no switchport
Ruijie(config-GigabitEthernet 0/1)# ip address 5.5.5.1 255.255.255.0
Ruijie(config-GigabitEthernet 0/1)# ip pim sparse-mode
Ruijie(config-GigabitEthernet 0/1)# exit
Ruijie(config)# interface gigabitethernet 0/2
```

# "no switchport" command is used to switch the port mode to "Routed Port" mode on switch products, and is not applicable to the router. Therefore, you don't need to execute this command on router products.

```
Ruijie(config-GigabitEthernet 0/2)# no switchport
Ruijie(config-GigabitEthernet 0/2)# ip address 4.4.4.2 255.255.255.0
Ruijie(config-GigabitEthernet 0/2)# ip pim sparse-mode
Ruijie(config-GigabitEthernet 0/2)# exit
Ruijie(config)# interface loopback 0
Ruijie(config-Loopback 0)# ip address 20.1.1.2 255.255.255.255
Ruijie(config-Loopback 0)# ip pim sparse-mode
Ruijie(config-Loopback 0)# end
```

# Configure BGP protocol to establish IBGP session with PE2 and EBGP session with ASBR1, and configure MDT address family

```
Ruijie# configure terminal
Ruijie(config)# router bgp 2
Ruijie(config-router)# neighbor 20.1.1.1 remote-as 2
Ruijie(config-router)# neighbor 20.1.1.1 update-source loopback 0
Ruijie(config-router)# neighbor 4.4.4.1 remote-as 1
Ruijie(config-router)# address-family ipv4 mdt
Ruijie(config-router-af)# neighbor 20.1.1.1 activate
Ruijie(config-router-af)# neighbor 4.4.4.1 activate
```

■ PE2:

The configuration steps are similar to that of PE1.

■ Verify configurations:

# Execute "**show ip pim mdt**" command on PE1 to display the MDT established.

```
Ruijie# show ip pim mdt
* implies group is the MDT default group
  MDT Group      Interface   Source                VRF
* 232.1.1.1      Tunnel0     Loopback0             vpn1
```

# Execute "**show ip pim sparse-mode neighbor**" command on PE1 to display the neighbors established under VRF.

```
Ruijie# show ip pim sparse-mode vrf vpn1 neighbor
Neighbor    Interface           Uptime/Expires     Ver   DR
Address                                                  Priority/Mode
192.168.0.1 Gigabitethernet 0/1 02:34:20/00:01:31  v2    1 / P
20.1.1.1     Tunnel0             01:55:19/00:01:37  v2    1 / P
```

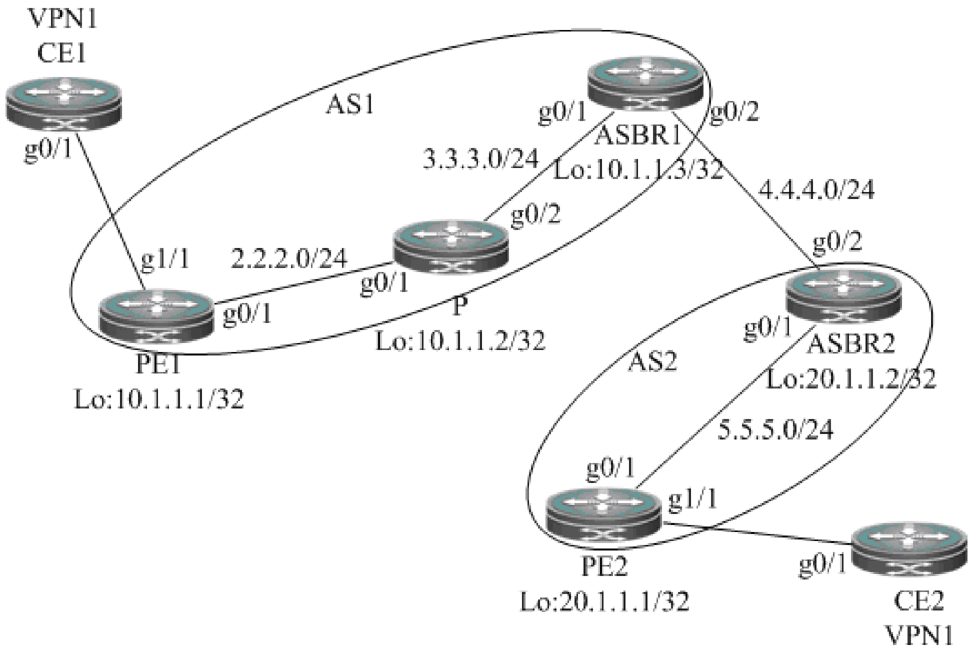
# Execute "**show ip pim sparse-mode mroute proxy**" command on PE1 to display RPF Vector information.

```
Ruijie# show ip pim sparse-mode mroute proxy
(20.1.1.1, 232.1.1.1)
  Proxy                    Assigner        Origin    Uptime/Expire
  1:100/10.1.1.3           0.0.0.0         BGP MDT   00:09:25/stopped
```

Example of Multicast VPN Configuration in OptionC when Using PIM-SSM

Requirement: BGP/MPLS L3VPN has been configured according to the scheme of OptionC.

Figure 30 Topology of multi-AS multicast VPN (OptionC)



Configuration steps:

- CE1:

The configuration steps are similar to that of CE1 in the scheme of single-AS multicast VPN.

- CE2:

The configuration steps are similar to that of CE2 in the scheme of single-AS multicast VPN.

- PE1:

# Configure VRF

Create a VRF of "vpn1", define RD value and RT value and configure Default-MDT. The group address of Default-MDT must fall within the scope of PIM-SSM.

```
Ruijie# configure terminal
Ruijie(config)# ip vrf vpn1
Ruijie(config-vrf)# rd 1:100
Ruijie(config-vrf)# route-target both 1:100
Ruijie(config-vrf)# mdt default 232.1.1.1
Ruijie(config-vrf)# end
```

# Configure private network multicast routing

```
Ruijie# configure terminal
Ruijie(config)# ip multicast-routing vrf vpn1
Ruijie(config)# ip pim vrf vpn1 rp-address 192.168.0.1
Ruijie(config)# interface gigabitethernet 1/1
```

# "no switchport" command is used to switch the port mode to "Routed Port" mode on switch products, and is not applicable to the router. Therefore, you don't need to execute this command on router products.

```
Ruijie(config-GigabitEthernet 1/1)# no switchport
Ruijie(config-GigabitEthernet 1/1)# ip vrf forwarding vpn1
Ruijie(config-GigabitEthernet 1/1)# ip address 192.168.0.2 255.255.255.0
Ruijie(config-GigabitEthernet 1/1)# ip pim sparse-mode
Ruijie(config-GigabitEthernet 1/1)# end
```

# Configure public network multicast routing

```
Ruijie# configure terminal
Ruijie(config)# ip multicast-routing
Ruijie(config)# ip multicast rpf proxy vector
Ruijie(config)# ip pim ssm default
Ruijie(config)# interface gigabitethernet 0/1
```

# "no switchport" command is used to switch the port mode to "Routed Port" mode on switch products, and is not applicable to the router. Therefore, you don't need to execute this command on router products.

```
Ruijie(config-GigabitEthernet 0/1)# no switchport
Ruijie(config-GigabitEthernet 0/1)# ip address 2.2.2.1 255.255.255.0
Ruijie(config-GigabitEthernet 0/1)# ip pim sparse-mode
Ruijie(config-GigabitEthernet 0/1)# exit
Ruijie(config)# interface loopback 0
Ruijie(config-Loopback 0)# ip address 10.1.1.1 255.255.255.255
Ruijie(config-Loopback 0)# ip pim sparse-mode
Ruijie(config-Loopback 0)# end
```

# Configure BGP protocol to establish mhop-EBGP session with PE2 and configure MDT address family

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 20.1.1.1 remote-as 2
Ruijie(config-router)# neighbor 20.1.1.1 ebgp-multihop 255
Ruijie(config-router)# neighbor 20.1.1.1 update-source loopback 0
Ruijie(config-router)# address-family ipv4 mdt
Ruijie(config-router-af)# neighbor 20.1.1.1 activate
Ruijie(config-router-af)# end
```

- P:

The configuration steps are similar to that of P in the scheme of single-AS multicast VPN.

- ASBR1:

# Configure public network multicast routing

```
Ruijie# configure terminal
Ruijie(config)# ip multicast-routing
Ruijie(config)# interface gigabitethernet 0/1
```

# "no switchport" command is used to switch the port mode to "Routed Port" mode on switch products, and is not applicable to the router. Therefore, you don't need to execute this command on router products.

```
Ruijie(config-GigabitEthernet 0/1)# no switchport
Ruijie(config-GigabitEthernet 0/1)# ip address 3.3.3.2 255.255.255.0
Ruijie(config-GigabitEthernet 0/1)# ip pim sparse-mode
Ruijie(config-GigabitEthernet 0/1)# exit
Ruijie(config)# interface gigabitethernet 0/2
```

# "no switchport" command is used to switch the port mode to "Routed Port" mode on switch products, and is not applicable to the router. Therefore, you don't need to execute this command on router products.

```
Ruijie(config-GigabitEthernet 0/2)# no switchport
Ruijie(config-GigabitEthernet 0/2)# ip address 4.4.4.1 255.255.255.0
Ruijie(config-GigabitEthernet 0/2)# ip pim sparse-mode
Ruijie(config-GigabitEthernet 0/2)# exit
Ruijie(config)# interface loopback 0
Ruijie(config-Loopback 0)# ip address 10.1.1.3 255.255.255.255
Ruijie(config-Loopback 0)# ip pim sparse-mode
Ruijie(config-Loopback 0)# end
```

# Execute BGP/MPLS L3VPN (OptionC) related configurations on ASBR; there is no need to configure MDT address family.

- ASBR2:

The configuration steps are similar to that of ASBR1.

- PE2:

The configuration steps are similar to that of PE1.

- Verify configurations:

# Execute "**show ip pim mdt**" command on PE1 to display the MDT established.

```
Ruijie# show ip pim mdt
* implies group is the MDT default group
  MDT Group      Interface   Source               VRF
* 232.1.1.1      Tunnel0     Loopback0            vpn1
```

# Execute "**show ip pim sparse-mode neighbor**" command on PE1 to display the neighbors established under VRF.

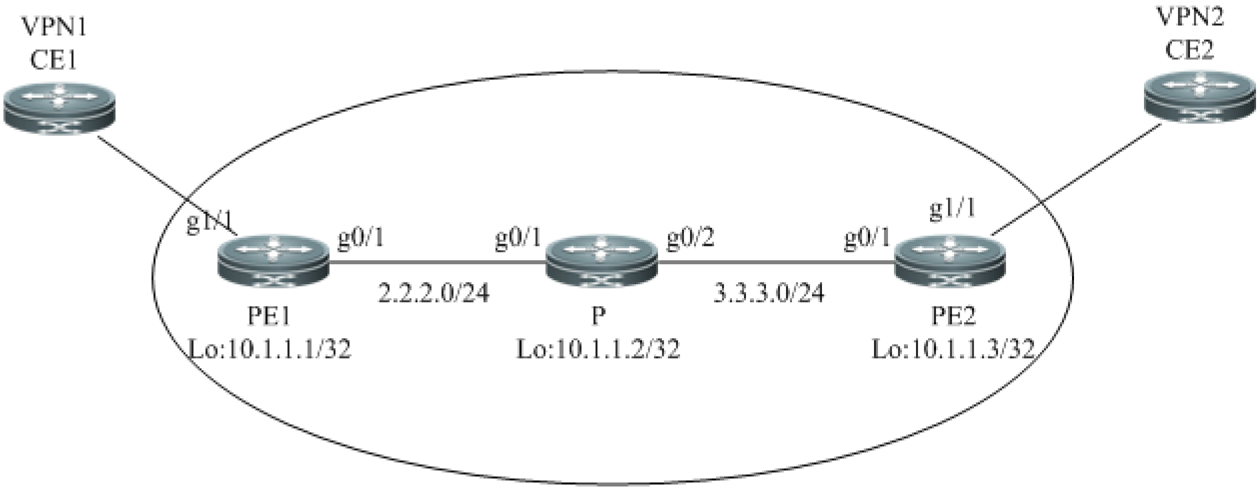
```
Ruijie# show ip pim sparse-mode vrf vpn1 neighbor
Neighbor    Interface          Uptime/Expires     Ver   DR
Address                                                Priority/Mode
192.168.0.1 Gigabitethernet 0/1 02:34:20/00:01:31  v2    1 / P
```

```
20.1.1.1     Tunnel0            01:55:19/00:01:37  v2    1 / P
```

Extranet-MVPN Configuration Instance of Configuring the Receiver MVRF on the Ingress PE

Prerequisite: Multicast VPN has been configured in single AS mode. Extranet-MVPN configuration in the cross-AS topology is the same as that in the single AS topology.

Figure 31 Topology of configuring the receiver MVRF on the ingress PE



Procedure:

- CE1:

# Configure the multicast route.

```
Ruijie# configure terminal
Ruijie(config)# ip multicast-routing
Ruijie(config)# ip pim rp-address 192.168.0.1
Ruijie(config)# interface gigabitethernet 0/1
Ruijie(config-GigabitEthernet 0/1)# ip address 192.168.0.1 255.255.255.0
Ruijie(config-GigabitEthernet 0/1)# ip pim sparse-mode
```

- PE1:

# Configure the VRF.

Configure an MVRF (vpn1), define the RD value and the RT value, and configure Default-MDT.

```
Ruijie# configure terminal
Ruijie(config)# ip vrf vpn1
Ruijie(config-vrf)# rd 1:100
Ruijie(config-vrf)# route-target both 1:100
Ruijie(config-vrf)# mdt default 239.1.1.1
Ruijie(config-vrf)# end
```

Configure a receiver MVRF (vpn2), import the vpn1 route, and configure Default-MDT.

```
Ruijie# configure terminal
```

```
Ruijie(config)# ip vrf vpn2
Ruijie(config-vrf)# rd 2:200
Ruijie(config-vrf)# route-target import 1:100
Ruijie(config-vrf)# mdt default 239.2.2.2
Ruijie(config-vrf)# end
```

# Configure the multicast route on a private network.

```
Ruijie# configure terminal
Ruijie(config)# ip multicast-routing vrf vpn1
Ruijie(config)# ip multicast-routing vrf vpn2
Ruijie(config)# ip pim vrf vpn1 rp-address 192.168.0.1
Ruijie(config)# ip pim vrf vpn2 rp-address 192.168.0.1
Ruijie(config)# interface gigabitethernet 1/1
```

# The **no switchport** command is used to switch the port mode to **Routed Port** on a switch and is inapplicable to a router. Therefore, it does not need to be run on a router.

```
Ruijie(config-GigabitEthernet 1/1)# no switchport
Ruijie(config-GigabitEthernet 1/1)# ip vrf forwarding vpn1
Ruijie(config-GigabitEthernet 1/1)# ip address 192.168.0.2 255.255.255.0
Ruijie(config-GigabitEthernet 1/1)# ip pim sparse-mode
Ruijie(config-GigabitEthernet 1/1)# end
```

# Configure the multicast route on a public network.

```
Ruijie# configure terminal
Ruijie(config)# ip multicast-routing
Ruijie(config)# ip pim rp-address 10.1.1.2
Ruijie(config)# interface gigabitethernet 0/1
```

# The no switchport command is used to switch the port mode to Routed Port and is inapplicable to a router. Therefore, it does not need to be run on a router.

```
Ruijie(config-GigabitEthernet 0/1)# no switchport
Ruijie(config-GigabitEthernet 0/1)# ip address 2.2.2.1 255.255.255.0
Ruijie(config-GigabitEthernet 0/1)# ip pim sparse-mode
Ruijie(config-GigabitEthernet 0/1)# exit
Ruijie(config)# interface loopback 0
Ruijie(config-Loopback 0)# ip address 10.1.1.1 255.255.255.255
Ruijie(config-Loopback 0)# ip pim sparse-mode
Ruijie(config-Loopback 0)# end
```

# Configure BGP, establish the IBGP session with PE2, and configure the VPNv4 address family.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 10.1.1.3 remote-as 1
Ruijie(config-router)# neighbor 10.1.1.3 update-source loopback 0
Ruijie(config-router)# address-family vpnv4
Ruijie(config-router-af)# neighbor 10.1.1.3 activate
Ruijie(config-router-af)# neighbor 10.1.1.3 send-community extended
Ruijie(config-router-af)# exit
Ruijie(config-router)# address-family ipv4 mdt
Ruijie(config-router-af)# neighbor 10.1.1.3 activate
Ruijie(config-router-af)# neighbor 10.1.1.3 send-community extended
Ruijie(config-router-af)# end
```

- P:

The procedure is similar to the procedure for configuring P on the single-AS multicast VPN.

- PE2:

# Configure the VRF.

Configure an MVRF (vpn1), define the RD value and the RT value, import the vpn1 route, and configure Default-MDT.

```
Ruijie# configure terminal
Ruijie(config)# ip vrf vpn2
Ruijie(config-vrf)# rd 2:200
Ruijie(config-vrf)# route-target both 2:200
Ruijie(config-vrf)# route-target import 1:100
Ruijie(config-vrf)# mdt default 239.2.2.2
Ruijie(config-vrf)# end
```

# Configure the multicast route on a private network.

```
Ruijie# configure terminal
Ruijie(config)# ip multicast-routing vrf vpn2
Ruijie(config)# ip pim vrf vpn2 rp-address 192.168.0.1
Ruijie(config)# interface gigabitethernet 1/1
```

# The **no switchport** command is used to switch the port mode to **Routed Port** on a switch and is inapplicable to a router. Therefore, it does not need to be run on a router.

```
Ruijie(config-GigabitEthernet 1/1)# no switchport
Ruijie(config-GigabitEthernet 1/1)# ip vrf forwarding vpn2
Ruijie(config-GigabitEthernet 1/1)# ip address 192.168.1.2 255.255.255.0
Ruijie(config-GigabitEthernet 1/1)# ip pim sparse-mode
Ruijie(config-GigabitEthernet 1/1)# end
```

# Configure the multicast route on a public network.

```
Ruijie# configure terminal
Ruijie(config)# ip multicast-routing
Ruijie(config)# ip pim rp-address 10.1.1.2
Ruijie(config)# interface gigabitethernet 0/1
```

# The **no switchport** command is used to switch the port mode to **Routed Port** and is inapplicable to a router. Therefore, it does not need to be run on a router.

```
Ruijie(config-GigabitEthernet 0/1)# no switchport
Ruijie(config-GigabitEthernet 0/1)# ip address 3.3.3.3 255.255.255.0
Ruijie(config-GigabitEthernet 0/1)# ip pim sparse-mode
Ruijie(config-GigabitEthernet 0/1)# exit
Ruijie(config)# interface loopback 0
Ruijie(config-Loopback 0)# ip address 10.1.1.3 255.255.255.255
Ruijie(config-Loopback 0)# ip pim sparse-mode
Ruijie(config-Loopback 0)# end
```

# Configure BGP, establish the IBGP session with PE2, and configure the VPNv4 address family.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 10.1.1.1 remote-as 1
Ruijie(config-router)# neighbor 10.1.1.1 update-source loopback 0
Ruijie(config-router)# address-family vpnv4
Ruijie(config-router-af)# neighbor 10.1.1.1 activate
Ruijie(config-router-af)# neighbor 10.1.1.1 send-community extended
Ruijie(config-router-af)# exit
Ruijie(config-router)# address-family ipv4 mdt
Ruijie(config-router-af)# neighbor 10.1.1.1 activate
Ruijie(config-router-af)# neighbor 10.1.1.1 send-community extended
Ruijie(config-router-af)# end
```

- CE2:

# Configure the multicast route.

```
Ruijie# configure terminal
Ruijie(config)# ip multicast-routing
Ruijie(config)# ip pim rp-address 192.168.0.1
Ruijie(config)# interface gigabitethernet 0/1
Ruijie(config-GigabitEthernet 0/1)# ip address 192.168.1.1 255.255.255.0
Ruijie(config-GigabitEthernet 0/1)# ip pim sparse-mode
```

- Verify the configuration result.

# Run the **show ip mroute** command on PE1 to check the establishment of each VPN multicast forwarding table.

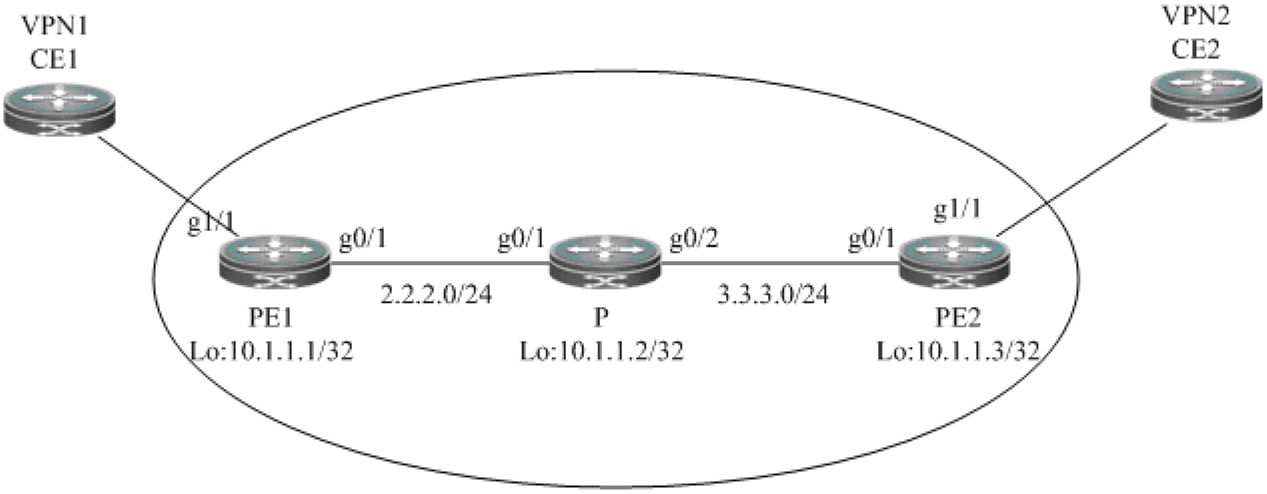
```
Ruijie# show ip mroute vrf vpn1

IP Multicast Routing Table
Flags: I - Immediate Stat, T - Timed Stat, F - Forwarder installed, R - RPT, S - SPT
Timers: Uptime/Stat Expiry
Interface State: Interface (TTL)

(1.1.1.1, 224.1.1.1), uptime 00:00:32, stat expires 00:02:58
Owner PIMSM, Flags: TFR
  Incoming interface: GigabitEthernet 1/1
  Outgoing interface list:

Extranet receivers in vrf vpn2:
```

```
(1.1.1.1, 224.1.1.1), 00:00:32/stopped

Ruijie# show ip mroute vrf vpn2

IP Multicast Routing Table
Flags: I - Immediate Stat, T - Timed Stat, F - Forwarder installed, R - RPT, S - SPT
Timers: Uptime/Stat Expiry
Interface State: Interface (TTL)

(1.1.1.1, 224.1.1.1), uptime 00:00:32, stat expires 00:02:58
Owner PIMSM, Flags: TFR
  Incoming interface: using vrf vpn1
  Outgoing interface list:
Tunnel 0 (1)
```

Extranet-MVPN Configuration Instance of Configuring the Source MVRF on the Egress PE

Prerequisite: Multicast VPN has been configured in single AS mode. Extranet-MVPN configuration in the cross-AS topology is the same as that in the single AS topology.

Figure 32 Topology of configuring the receiver MVRF on the ingress PE



Procedure:

- CE1:

# Configure the multicast route.

```
Ruijie# configure terminal
Ruijie(config)# ip multicast-routing
Ruijie(config)# ip pim rp-address 192.168.0.1
Ruijie(config)# interface gigabitethernet 0/1
Ruijie(config-GigabitEthernet 0/1)# ip address 192.168.0.1 255.255.255.0
Ruijie(config-GigabitEthernet 0/1)# ip pim sparse-mode
```

- PE1:

# Configure the VRF.

Configure an MVRF (vpn1), define the RD value and the RT value, and configure Default-MDT.

```
Ruijie# configure terminal
Ruijie(config)# ip vrf vpn1
Ruijie(config-vrf)# rd 1:100
Ruijie(config-vrf)# route-target both 1:100
Ruijie(config-vrf)# mdt default 239.1.1.1
Ruijie(config-vrf)# end
```

# Configure the multicast route on a private network.

```
Ruijie# configure terminal
Ruijie(config)# ip multicast-routing vrf vpn1
Ruijie(config)# ip pim vrf vpn1 rp-address 192.168.0.1
Ruijie(config)# interface gigabitethernet 1/1
```

# The **no switchport** command is used to switch the port mode to **Routed Port** on a switch and is inapplicable to a router. Therefore, it does not need to be run on a router.

```
Ruijie(config-GigabitEthernet 1/1)# no switchport
Ruijie(config-GigabitEthernet 1/1)# ip vrf forwarding vpn1
Ruijie(config-GigabitEthernet 1/1)# ip address 192.168.0.2 255.255.255.0
Ruijie(config-GigabitEthernet 1/1)# ip pim sparse-mode
Ruijie(config-GigabitEthernet 1/1)# end
```

# Configure the multicast route on a public network.

```
Ruijie# configure terminal
Ruijie(config)# ip multicast-routing
Ruijie(config)# ip pim rp-address 10.1.1.2
Ruijie(config)# interface gigabitethernet 0/1
```

# The no switchport command is used to switch the port mode to Routed Port on a switch and is inapplicable to a router. Therefore, it does not need to be run on a router.

```
Ruijie(config-GigabitEthernet 0/1)# no switchport
Ruijie(config-GigabitEthernet 0/1)# ip address 2.2.2.1 255.255.255.0
Ruijie(config-GigabitEthernet 0/1)# ip pim sparse-mode
Ruijie(config-GigabitEthernet 0/1)# exit
Ruijie(config)# interface loopback 0
Ruijie(config-Loopback 0)# ip address 10.1.1.1 255.255.255.255
Ruijie(config-Loopback 0)# ip pim sparse-mode
Ruijie(config-Loopback 0)# end
```

# Configure BGP, establish the IBGP session with PE2, and configure the VPNv4 address family.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 10.1.1.3 remote-as 1
Ruijie(config-router)# neighbor 10.1.1.3 update-source loopback 0
Ruijie(config-router)# address-family vpnv4
Ruijie(config-router-af)# neighbor 10.1.1.3 activate
Ruijie(config-router-af)# neighbor 10.1.1.3 send-community extended
Ruijie(config-router-af)# end
```

- P:

The procedure is similar to the procedure for configuring P on the single-AS multicast VPN.

- PE2:

# Configure the VRF.

Configure an MVRF (vpn2), define the RD value and the RT value, and configure Default-MDT.

```
Ruijie# configure terminal
Ruijie(config)# ip vrf vpn2
Ruijie(config-vrf)# rd 2:200
Ruijie(config-vrf)# route-target both 2:200
Ruijie(config-vrf)# mdt default 239.2.2.2
Ruijie(config-vrf)# end
```

Configure a source MVRF (vpn1), define the RD value and the RT value, and configure Default-MDT.

```
Ruijie# configure terminal
Ruijie(config)# ip vrf vpn1
Ruijie(config-vrf)# rd 1:100
Ruijie(config-vrf)# route-target both 1:100
Ruijie(config-vrf)# mdt default 239.1.1.1
Ruijie(config-vrf)# end
```

# Specify the VRF to be selected by the static multicast route.

```
Ruijie# configure terminal
Ruijie(config)# ip mroute vrf vpn2 192.168.0.0 255.255.255.0 fallback-lookup vrf vpn1
```

# Configure the multicast route on a private network.

```
Ruijie# configure terminal
Ruijie(config)# ip multicast-routing vrf vpn1
Ruijie(config)# ip multicast-routing vrf vpn2
Ruijie(config)# ip pim vrf vpn1 rp-address 192.168.0.1
Ruijie(config)# ip pim vrf vpn2 rp-address 192.168.0.1
Ruijie(config)# interface gigabitethernet 1/1
```

# The **no switchport** command is used to switch the port mode to **Routed Port** on a switch and is inapplicable to a router. Therefore, it does not need to be run on a router.

```
Ruijie(config-GigabitEthernet 1/1)# no switchport
Ruijie(config-GigabitEthernet 1/1)# ip vrf forwarding vpn2
Ruijie(config-GigabitEthernet 1/1)# ip address 192.168.1.2 255.255.255.0
Ruijie(config-GigabitEthernet 1/1)# ip pim sparse-mode
Ruijie(config-GigabitEthernet 1/1)# end
```

# Configure the multicast route on a public network.

```
Ruijie# configure terminal
Ruijie(config)# ip multicast-routing
Ruijie(config)# ip pim rp-address 10.1.1.2
Ruijie(config)# interface gigabitethernet 0/1
```

# The no switchport command is used to switch the port mode to Routed Port on a switch and is inapplicable to a router. Therefore, it does not need to be run on a router.

```
Ruijie(config-GigabitEthernet 0/1)# no switchport
Ruijie(config-GigabitEthernet 0/1)# ip address 3.3.3.3 255.255.255.0
Ruijie(config-GigabitEthernet 0/1)# ip pim sparse-mode
Ruijie(config-GigabitEthernet 0/1)# exit
Ruijie(config)# interface loopback 0
Ruijie(config-Loopback 0)# ip address 10.1.1.3 255.255.255.255
Ruijie(config-Loopback 0)# ip pim sparse-mode
Ruijie(config-Loopback 0)# end
```

# Configure BGP, establish the IBGP session with PE2, and configure the VPNv4 address family.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 10.1.1.1 remote-as 1
Ruijie(config-router)# neighbor 10.1.1.1 update-source loopback 0
Ruijie(config-router)# address-family vpnv4
Ruijie(config-router-af)# neighbor 10.1.1.1 activate
Ruijie(config-router-af)# neighbor 10.1.1.1 send-community extended
Ruijie(config-router-af)# end
```

- CE2:

# Configure the multicast route.

```
Ruijie# configure terminal
Ruijie(config)# ip multicast-routing
Ruijie(config)# ip pim rp-address 192.168.0.1
Ruijie(config)# interface gigabitethernet 0/1
Ruijie(config-GigabitEthernet 0/1)# ip address 192.168.1.1 255.255.255.0
Ruijie(config-GigabitEthernet 0/1)# ip pim sparse-mode
```

MPLS Configuration

Configuring MPLS

Note The term “router” and router icons in this chapter refer to routers in a generic sense and layer-3 switches running routing protocols.

Understanding MPLS

Multiprotocol Label Switching (MPLS) supports multiple network-layer protocols such as IP, IPv6, and IPX and is compatible with multiple link-layer technologies including ATM, frame relay, Ethernet, and PPP. MPLS forwards packets based on labels attached to them. It works at both the connectionless control plane and the connection-oriented data plane, thereby introducing connection-oriented attributes to connectionless IP networks. The MPLS technology was first introduced to enhance the forwarding rate of routing devices. However, with the development of hardware technologies and network processors, this competitive edge gradually loses its appeal. Because MPLS combines Layer 2 switching with Layer 3 routing technologies, it has unprecedented edges over other technologies in terms of addressing issues of virtual private networks (VPNs) and traffic engineering (TE). MPLS VPN is increasingly favored by carriers to address interconnection problems between companies and to provide various new services. It has already become an important means for carriers to provide value-added services over IP networks. At the same time, the MPLS TE technology also turns into a major method to reduce congestion and guarantee quality of service (QoS) on IP networks by managing network traffic. Therefore, the MPLS technology receives more and more attention and the MPLS applications gradually shift to MPLS VPN and TE applications.

- Basic Concept
- Different from incoming label map (ILM), FEC-to-NHLFE (FTN) maps each forwarding equivalence class (FEC) to a series of next hop label forwarding entries (NHLFEs) which indicates there are multiple paths. An FTN table is used when a label edge router (LER) encapsulates a label in an unlabeled packet before forwarding the packet.
- Label
- Label Distribution Protocol
- MPLS Network
- An MPLS network comprises two basic components: Label Switching Routers (LSRs) and LERs. An LSR is located at the core of the MPLS network and runs the LDP signaling protocol to forward labeled packets. An LER classifies and labels incoming packets into FECs and encapsulates the labeled packets as MPLS packets for forwarding. The LER also removes the labels from outgoing MPLS packets and restores these packets to the original packets. On the MPLS network, packets with labels are forwarded along the label switched path (LSP) set up through LDP.

The MPLS architecture is divided into two parts: the forwarding unit (data plane) and control unit (control plane). The former forwards a packet by searching the label forwarding information base (LFIB) based on the label carried by the packet whereas the latter is responsible for creating and maintaining the LFIB between the connected MPLS nodes. Each MPLS node must run one or more routing protocols (including static routes) to exchange routing information with other MPLS nodes on the MPLS network. Therefore, in effect, each MPLS node is an IP router on the control plane. Similar to a conventional IP router, an MPLS node also uses unicast routing protocols (including static routes) to create and maintain a routing table. The difference is that the traditional router uses the routing table to create a forwarding table and the

MPLS node uses the routing table to exchange label binding information between each destination subnet and neighboring MPLS nodes. The protocol responsible for exchanging label binding information is LDP.

- MPLS Forwarding Behaviors
- The MPLS forwarding process is as follows (IP routing for example):

1) All LSRs (including LERs) run routing protocols such as Open Shortest Path First (OSPF) and Intermediate System to Intermediate System (IS-IS) to establish IP routing tables on LSRs and LERs.
2) LDP creates an LSP based on the IP routing tables.
3) The ingress LER receives an IP packet, analyzes the IP header, associates it with an FEC, labels the IP header with L1, and sends the labeled packet to the next-hop LSR along the LSP.
4) After receiving the packet, the next-hop LSR searches the next LSP based on the label on the stack top, replaces the label with a new label, and sends the packet to the next LSR of the LSP.
5) The subsequent LSRs repeat as step 4.
6) When the last but one LSR receives the labeled packet and looks up the LFIB, if the egress label is found to be an implicit label, it pops the label and forwards the pure IP packet to the last hop LSR; and if the egress label is an explicit null one, the LSR pops the label and forwards the packet by looking up the IP forwarding table based on the IP header..
7) If the last but one LSR pops the label, the egress LER receives the original IP packet and forwards the packet according to the IP routing table.

- LSP Establishment and Loop Detection
- MPLS Applications

Basic Concepts

- MPLS Node

MPLS-enabled nodes can identify MPLS signaling protocols (control protocols), support one or more Layer 3 routing protocols (including static routes), and forward packets based on MPLS labels. Generally, an MPLS node can forward original Layer 3 packets (such as IP packets).

- FEC

An FEC refers to a set of data packets that are forwarded in the same way, such as data packets that have the same prefix in their destination addresses. The FEC supports different classification methods for different applications. For example, the FEC classifies IP unicast routes based on the address prefixes. That is, one route corresponds to one FEC. All the packets in the same FEC are equally handled on the MPLS network.

- LSR

As a core device on an MPLS network, the LSR provides label switching and distribution functions. As specified in RFC 3031 for MPLS system, an LSR is also an MPLS node which is capable of forwarding original Layer 3 packets (such as IPv4 andIPv6 packets). Since MPLS can forward normal IP packets, LSRs also have this capability.

- LER

Located on the edge of an MPLS network, an LER classifies incoming traffic into different FECs, requests labels for these FECs, and restores outgoing traffic to the original packets by popping labels. The LER thus provides traffic classification, label mapping, and label removal functions.

- LSP

An FEC data stream is assigned with specific labels on different nodes and transmitted along the nodes according to the switching of assigned labels. The path where the data stream travels is an LSP. It is a collection of multiple LSRs. Therefore, an LSP can be considered as a tunnel that traverses the MPLS core network.

- NHLFE

An NHLFE table is used to store the next-hop information about MPLS packets. Typically, a next hop label forwarding entry contains the following information:

1) Next hop of data packets
2) Link layer encapsulation to use when forwarding data packets
3) Coding scheme for the label stack when forwarding data packets
4) Operations on the label stack of data packets, including:
    a) Replacing the label at the top of the label stack with a new label
    b) Popping the label off the stack top
    c) Pushing one or more labels
    d) Replacing the label at the top of the label stack with a new label and pushing one or more new labels on the label stack

- ILM

An ILM table is a label forwarding table where each incoming label is mapped to a series of NHLFEs (multiple NHLFEs indicate multiple paths). The ILM is used when an LSR receives and forwards MPLS packets with labels.

- FTN

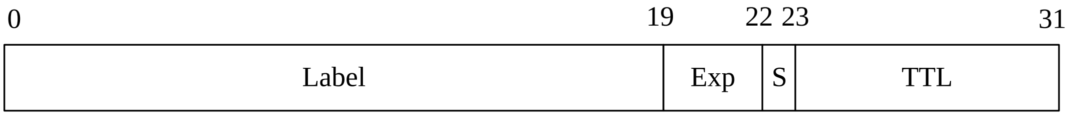
Different from ILM, FTN maps each FEC to a series of NHLFEs (multiple NHLFEs indicate multiple paths). An FTN table is used when an LER encapsulates labels to an unlabeled packet before forwarding the packet.

Label

A label is a short identifier with fixed length and of local significance. The label is distributed and transmitted only between two adjacent LSRs. As a result, it is valid only between the two LSRs. One label identifies one FEC. When arriving at the MPLS ingress, packets are classified into different FECs according to certain rules. Based on the FECs, the packets are encapsulated with different labels and then forwarded on the MPLS network based on the labels.

Label Format

Figure 1 MPLS label format



As shown in Figure-1, a label consists of four fields, which is described as follows:

- Label field

The label field is a 20-bit label value. The label value is an index for the label forwarding table. The Internal Engineering Task Force (IETF) classifies 0 to 15 as reserved labels and predefines their meanings:

Reserved Label	**Description**

0	IPv4 explicit null label. As specified in RFC 3032, label 0 must be placed at the bottom of the label stack, which means that the label must be popped before packets are forwarded according to destination IP addresses. RFC 4182 modifies the description of label 0 in RFC 3032. In RFC 4182, label 0 is popped directly upon receipt of a label 0 packet. The subsequent forwarding behavior is determined by the content behind the label 0. If another label follows, the packet is forwarded according to the label; if the packet is an IPv4 packet, it is forwarded based on the destination IP address.
1	Router alert label. This label is not allowed to be placed at the bottom of the label stack. When a receiving packet carries a router alert label, it must be sent to the local software module for processing. The actual forwarding of the packet must be based on the label behind the router alert label. Before the packet is forwarded, however, the router alert label must be pushed to the label stack again. This option is similar to the Router Alert Option of IP packets. You can use this option to configure each LSR to check MPLS packets.
2	IPv6 explicit null label. As specified in RFC 3032, label 2 must be placed at the bottom of the label stack. This means that the label must be popped before the packet is forwarded according to the destination IP address. RFC 4182 modifies the description of label 2 in RFC 3032. In RFC 4182, label 2 is popped directly upon receipt of a label 2 packet. The subsequent forwarding behavior is based on the content behind the label2. If another label follows, the packet is forwarded according to the label; if it is an IPv6 packet, it is forwarded according to the destination IP address.
3	Implicit null label. This label can be distributed by LDP but can never be transmitted in the label stacks of MPLS packets. When an LSR exchanges MPLS packets, if the label on the top of the label stack has a value of 3, it can be popped but should not be replaced with another label. The implicit null label is used in the Penultimate Hop Popping (PHP) function.
4 to 15	Reserved by the IETF for future use.

- Exp field

The Exp field is currently used to store the QoS information about MPLS. This field contains 3 bits.

- S mark

The S mark field indicates the stack bottom. It contains one bit. If multiple labels exist, the S bit of the label at the stack bottom is set to **1** and the S bits of other labels are **0**. If only one label exists, the S bit is set to 1.

- TTL

Time to Live. This field contains 8 bits. It is similar to the TTL value in an IP packet header. When a label is first added to an IP packet, the TTL value can be copied from the TTL field (or Hop Limit of IPv6) of the IP packet header. The TTL value of the outer (stack top) label then decreases by one at every label switching. When MPLS runs on ATM links, the label encoding methods are different and no TTL field exists. For methods and solutions, refer to RFC 3032.

Label Stack

An MPLS packet can contain several labels to form a label stack. The label after the link layer header is the top label and the label before the IP header is the bottom label. An LSR exchanges labels based on the top label. When multiple labels exist, each label must be complete and contain 32 bits. A label stack supports multiple layers of labels to be carried in an MPLS packet. The purpose is to enable the MPLS technology to support hierarchical network systems and LSP tunnels.

Operation Methods of Labels

The following lists basic label operations on MPLS nodes:

- Push

Insert a label between the link layer header and the network layer header on an ingress LER or add a new label to the stack top of an MPLS packet on an intermediate LSR.

- Pop

Remove labels off packets on the egress LER to restore the IP packets or remove the top label on an intermediate LSR to reduce layers of a label stack.

- Swap

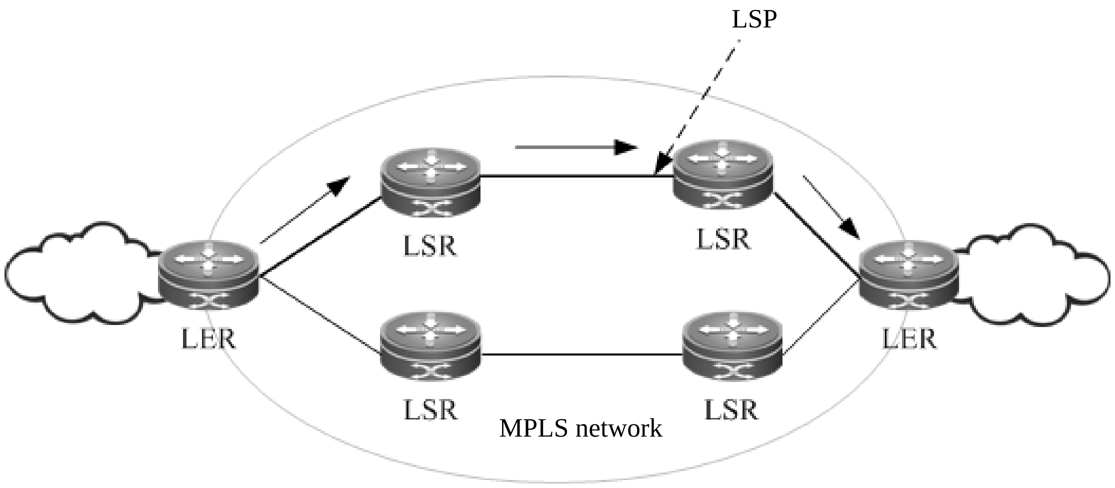
Replace the top label in the label stack of a packet based on the ILM during forwarding of the packet.

LDP

As an emerging network system, MPLS also has its own signaling protocols or "routing protocols". One of the basic concepts in the MPLS system is that two LSRs must reach consensus on the meaning of labels used for traffic transmission. This consensus is realized through a series of processes, that is, LDP. Through LDP, one LSR can notify the other LSR of the label binding. The MPLS system architecture does not assume that there is only a single LDP. Some MPLS systems use independent distribution protocols, such as LDP defined in RFC 3036 by the IETF; other MPLS systems support the distribution of labels by extending existing protocols in piggybacking mode, such as MP-BGP and RSVP. You can choose different LDPs for MPLS networks based on different application scenarios.

MPLS Network

Figure 2

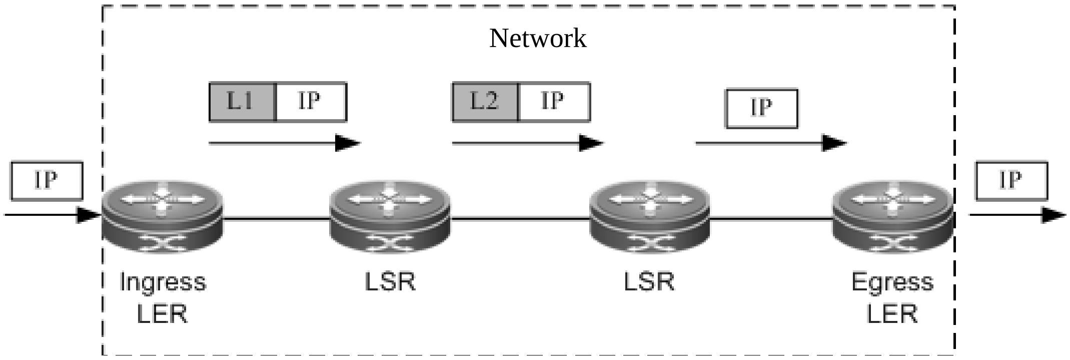


An MPLS network comprises two basic components: Label Switching Routers (LSRs) and LERs. An LSR is located at the core of the MPLS network and runs the LDP signaling protocol to forward labeled packets. An LER classifies and labels incoming packets into FECs and encapsulates the labeled packets as MPLS packets for forwarding. The LER also removes the labels from outgoing MPLS packets and restores these packets to the original packets. On the MPLS network, packets with labels are forwarded along the label switched path (LSP) set up through LDP.

The MPLS architecture is divided into two parts: the forwarding unit (data plane) and control unit (control plane). The former forwards a packet by searching the label forwarding information base (LFIB) based on the label carried by the packet whereas the latter is responsible for creating and maintaining the LFIB between the connected MPLS nodes. Each MPLS node must run one or more routing protocols (including static routes) to exchange routing information with other MPLS nodes on the MPLS network. Therefore, in effect, each MPLS node is an IP router on the control plane. Similar to a conventional IP router, an MPLS node also uses unicast routing protocols (including static routes) to create and maintain a routing table. The difference is that the traditional router uses the routing table to create a forwarding table and the MPLS node uses the routing table to exchange label binding information between each destination subnet and neighboring MPLS nodes. The protocol responsible for exchanging label binding information is LDP.

MPLS Forwarding Behaviors

Figure 3 Forwarding process of MPLS packets that support PHP



The following takes traditional IP routing services as an example to show the MPLS forwarding process:

- Enable traditional routing protocols (OSPF or IS-IS) on all LSRs (including LERs) and create IP routing tables on the LSRs and LERs.
- Set up an LDP LSP based on the IP routing table.
- Upon receipt of an IP packet, the ingress LER analyzes the IP packet header and maps it to an FEC. The ingress LER then adds the label L1 to the packet and sends the labeled packet to the next hop LSR along the LSP.
- The next-hop LSR receives the labeled packet, searches the LSP based on the label on the stack top, and then forwards the packet to the next-hop LSR along the LSP after replacing the label with a new one.
- The intermediate LSRs perform the same actions as step 4.
- Upon receipt of the labeled packet, the PHP LSR searches the label forwarding table and pops the label after learning that the outgoing label is the implicit null label 3. The PHP LSR then forwards the original IP packet to the last-hop LSR. If the outgoing label is the explicit null label, the PHP LSR pops the label and directly sends the original packet based on the routes of the IP header in the IP forwarding table.
- If the label is popped on the PHP LSR, the last-hop egress LER receives the original IP packet and forwards it according to the IP routing table.

LSP Establishment and Loop Detection

A virtual MPLS connection is an LSP. One FEC data stream is assigned with different labels on different MPLS nodes and forwarded according to the labels. The path that the data stream travels is an LSP that consists of a series of LSRs. Data streams of the same FEC pass through the same LSP.

- LSP Establishment
- LSP Loop Control

LSP Establishment

The LSP establishment is the process of binding an FEC to a label and notifying adjacent LSRs of the binding. This process is completed by LDP. RFC 3036 stipulates the protocol specifications of LDP, the interactive process of LSRs, and the message formats.

LDP detects adjacent LSRs by sending Hello messages periodically. The LDP Hello messages are encapsulated using the User Datagram Protocol (UDP) and use the well-known port 646 as the destination port. The destination address of these packets is 224.0.0.2, the multicast address of all routers in the subnet. The discovery of an adjacent LSR triggers the creation of LDP sessions. An LDP session is created in the following two steps:

- Establish a transmission connection. The connection is established after the completion of TCP three-way handshakes that do not require any interaction of LDP messages.
- Initialize the session. Both parties exchange their initialization information to negotiate and determine the LDP session parameters such as the label distribution mode, Keepalive duration, and the maximum length of Protocol Data Unit (PDU).

After the LDP session is created and both parties enter the operational state, the two parties can exchange label messages to distribute and manage labels, and create an LSP for each FEC.

During the process of LSP establishment, two label distribution modes are used: Downstream on Demand (DOD) and Downstream Unsolicited (DU). In DOD mode, an LSR responds to a label binding message only after it receives a label request from an adjacent LSR. In DU mode, the LSR voluntarily sends label binding messages to its adjacent LSRs without receiving any request.

During the LSP establishment, two label control methods are used: independent and ordered control. In independent control mode, each LSR announces to its adjacent devices the binding of labels and FECs at any required time. In independent DOD mode, one LSR can immediately answer an upstream label mapping request without waiting for the label mapping from the next hop device. In independent DU mode, one LSR can announce the label mapping of an FEC at any time deemed as proper for swapping the label of the FEC.

In ordered control mode, one LSR binds an FEC to a label and sends the binding upstream only when the FEC has the next-hop label mapping or the LSR is the egress of the FEC. Otherwise, the LSR does not bind the FEC to a label, or send the binding to an upstream LSR until receiving the label mapping of the FEC from a downstream LSR. In ordered control and DU mode, one LSR announces the label to an upstream LSR only when the LSR is the egress of the FEC or the LSR receives the label distributed by a downstream LSR. If the label distribution mode of the downstream LSR is DOD, the LSR, either in DOD or DU mode, passes on the label request from an upstream LSR to downstream devices.

LSP Loop Control

During the LSP setup process, the loop detection mechanism must be provided to ensure timely detection of any loops formed by the LSP. The maximum number of hops and path vector can be used to avoid LSP loops.

When the maximum number of hops is used, the message that transmits label binding information records the number of bypassing LSRs. The number increases by one after passing an LSR. If the number exceeds the specified maximum value, the system considers that a loop occurs and terminates the LSP.

When the path vector is used, the message that transmits label binding information records IDs of bypassing LSRs. The ID of an LSR is recorded to the vector table of the message after each LSR. Upon receipt of a label binding message, an LSR checks whether its ID is included in the vector table. If not, the LSR adds its ID to the record when distributing the message; if yes, the LSR considers that a loop occurs and terminates the LSP.

MPLS Applications

Thanks to the combination of Layer 2 switching and Layer 3 routing technologies, the MPLS technology improves the forwarding rate of packets. With the development of the Application-Specific Integrated Circuit (ASIC) technologies, the forwarding rate is no longer a bottleneck in network development. As a result, the edges of MPLS in enhancing forwarding rates are not remarkable. Due to the innate advantage of combining Layer 2 switching and Layer 3 routing technologies, however, MPLS still has unprecedented edges over other technologies in terms of VPNs and TE. In this context, MPLS receives more and more attention. The MPLS applications also gradually shift to the application areas of MPLS VPN and MPLS TE.

Configuring MPLS

Caution

1. LDP is a topology-driven protocol. To ensure normal operation of LDP, enable IPv4 routing protocols and ensure their normal operations.
2. To enable the router MPLS express forwarding function and improve the forwarding performance of routers, use the **ip ref** command in interface configuration mode.

Enabling MPLS Forwarding Globally

Use the **mpls ip** command to enable a device to support MPLS forwarding in configuration mode. By default, MPLS forwarding is disabled on a device. After MPLS forwarding is enabled, the device first forwards packets according to their labels. When the label forwarding fails, the device attempts to forward packets based on their IP addresses.

Use the **no mpls ip** command to disable MPLS forwarding.

Command	Function
Ruijie(config)# **mpls ip**	Enables MPLS forwarding globally.
Ruijie(config)# **no mpls ip**	Disables MPLS forwarding globally.

Caution

This command is not applicable to controlling switch chip forwarding.

Enabling LDP Globally

Use the **mpls router ldp** [*vrf-name*] command to enable LDP for a VRF instance in global configuration mode and enter LDP configuration mode.

Use the **no mpls router ldp** [*vrf-name*] command to disable LDP for a VRF instance.

Command	Function
Ruijie(config)# **mpls router ldp** [*vrf-name*]	Enables LDP for a VRF instance and enters LDP configuration mode.
Ruijie(config-mpls-router)# **ldp router-id interface loopback** *id* [**force**]	Configures the LDP router ID. The loopback address is generally used as the router ID.
Ruijie(config)# **no mpls router ldp** [*vrf-name*]	Disables LDP for a VRF instance.

Caution

1. After LDP is enabled globally, use the **mpls ip** command to enable LDP for an interface in interface configuration mode.
2. If *vrf-name* is not specified, LDP is globally enabled for all VRF instances.
3. You are required to specify the router ID for an LDP when enabling LDP.

Enabling Label Switching on an Interface

By default, interfaces do not forward MPLS packets. To enable MPLS forwarding on a device, use the **mpls ip** command in global configuration mode. To explicitly enable MPLS forwarding on a specified interface, use the **label-switching** command. When enabling the label forwarding function of an interface, adjust the maximum transmission unit (MTU) of the interface based on service types to facilitate the transmission of large packets.

Command	Function
Ruijie(config-if-*type ID*)# **label-switching**	Enables MPLS forwarding on an interface.
Ruijie(config-if-*type ID*)# **no label-switching**	Disables MPLS forwarding on an interface.

☑ To enable the MPLS express forwarding function of a router and to improve the forwarding performance, use the **ip ref** command in interface configuration mode

☑ If label forwarding is enabled on a switch on a VLAN, there must be only one member port. Otherwise, packets cannot be flooded on the VLAN.

☑ When unknown unicast packets or broadcast packets enter the switch through a port, these packets (including packets of VLANs with label forwarding disabled) cannot be forwarded if label forwarding is enabled on a VLAN where the port belongs to. You are advised to enable label forwarding on routing ports.

Note

After the MPLS packet forwarding is disabled on a public network interface, packet forwarding from the AC to the PW is not affected.

Enabling LDP on an Interface

After LDP is enabled globally, use the **mpls ip** command to enable LDP on an interface in interface configuration mode.

Command	Function
Ruijie(config-if-*type ID*)# **mpls ip**	Enables LDP on an interface.
Ruijie(config-if-*type ID*)# **no mpls ip**	Disables LDP on an interface.

Caution

After LDP is enabled in interface configuration mode, LDP does not take effect on an interface if the **mpls router ldp** command is not used in global configuration mode. To enable LDP on the interface, you must also use the **label-switching** command to enable MPLS forwarding on the interface.

Configuring MPLS MTU on an Interface (Optional)

By default, values of the MPLS MTU and the MTU are the same for an interface. The MPLS MTU determines whether MPLS packets must be fragmented during forwarding. The MPLS MTU indicates the overall length of MPLS encapsulation and encapsulated (such as IP) layers.

Use the **no mpls mtu** command to restore the default value of the MPLS MTU on an interface.

Command	Function
Ruijie(config-if-*type ID*)# **mpls mtu** *bytes*	Configures the MPLS MTU on an interface.
Ruijie(config-if-*type ID*)# **no mpls mtu**	Restores the default value of the MPLS MTU on an interface.

Caution

The MPLS MTU on an interface cannot exceed the actual size of packets transmitted on the interface. For switches that forward packets based on ASIC, this configuration is invalid. These switches forward packets based on the MTU configured on interfaces and directly discard packets that exceed the MTU rather than performing fragmentation. To adjust the MTU of an interface, use the **mtu** command in interface configuration mode. Fragmentation is supported by only process forwarding and router forwarding. In actual applications, you must adjust the MTU value to avoid performance degradation caused by fragmentation.

Fragmenting MPLS Packets (Optional)

By default, MPLS packets that exceed the MPLS MTU on an interface are fragmented as IP fragmentations. The fragmented IP packets are still encapsulated with the original labels and transmitted along the original LSP.

Use the **no mpls ip fragment** command to directly discard packets that must be fragmented.

Command	Function
Ruijie(config)# **no mpls ip fragment**	Directly discards MPLS packets that exceed the MPLS MTU on an interface.
Ruijie(config)# **mpls ip fragment**	Restores the default value to fragment packets that exceed the MPLS MTU on an interface.

Caution

This command is valid only for the encapsulated IP packets.

Handling ICMP Error Messages (Optional)

ICMP error messages (such as typical MPLS TTL timeout messages) generated during the forwarding of MPLS packets are forwarded along the LSP of the label stack by default. To provide different processing methods for MPLS packets with different numbers of labels, use the **mpls ip icmp-error pop** *labels* command. For ICMP error messages with the forwarding label stack not greater than the specified number of labels, they are forwarded through routes on the IP routing table where the FEC of the stack top label is stored. For ICMP error messages with the forwarding label stack greater than the specified number of labels, they are forwarded along the LSP of the original label stack.

Command	Function

Ruijie(config)# **mpls ip icmp-error pop** *labels*	Controls the handling of ICMP error messages generated by labeled MPLS packets.
Ruijie(config)# **no mpls ip icmp-error pop**	Restores the default value.

Configuring the MPLS TTL Replication Function (Optional)

There are two modes for handling the TTL of encapsulated and de-encapsulated IP (or MPLS) packets on an MPLS network:

- TTL replication mode: This is the default working mode. The procedure is as follows: When a label is pushed, the label TTL copies the TTL of the existing IP or MPLS header to the TTL field of the label. When a label is popped out, the TTL is copied back from the outer label to the inner IP packet or MPLS packet.
- TTL non-replication mode: In this mode, the TTL is not copied. The procedure is as follows: When a label is pushed, the TTL value of the label is directly set to 255. When a label is popped out, the original TTL value of the inner IP packet or MPLS packet is exposed and retained.

Use the **mpls ip ttl propagate** { **public** | **VPN** } command to configure the TTL replication function for packets sent and forwarded by a device.

Command	**Function**
Ruijie(config)# [**no**] **mpls ip ttl propagate public**	Enables or disables the TTL replication function for MPLS packets sent by the device.
Ruijie(config)# [**no**] **mpls ip ttl propagate VPN**	Enables or disables the TTL replication function for MPLS service packets forwarded by the device.

After the TTL replication function is enabled on an MPLS network, you can use the Tracert tool on a CE to track all the LSRs that the packets pass through in the MPLS domain. If the TTL non-replication mode is configured on PEs, the entire LSP of the packets is considered as only one hop.

Caution

After TTL replication is enabled, the TTL of the inner header is not copied but retained if it is smaller than the TTL of the outer header.

Verifying the MPLS Information

To view MPLS information and verify the configuration results, use the **show** commands in privileged mode.

- Displaying MPLS information

Display the utilization information about the label space and the interfaces enabled with MPLS. You can verify whether the configurations are accurate based on the information.

Command	**Function**
Ruijie# **show mpls summary**	Displays basic MPLS information.

- Displaying the MPLS forwarding table

Display contents of MPLS forwarding entries and MPLS forwarding entries added to an MPLS application protocol (such as LDP and MP-BGP).

Command	**Function**

Ruijie# **show mpls forwarding-table** [**summary** | [[*ip-address/mask* | **label** *label* | **interface** *interface-name* | **next-hop i***p-address*] | [**ftn** [**ip** | **vc**] | **ilm** [**ip** | **vc**]] | { **vrf** *vrf-name* | **global** } [**ftn** | **ilm**]] [**frr**] [**detail**]]	Displays the information about the MPLS forwarding table.

- Displaying the utilization of the label pool

Command	Function
Ruijie# **show mpls label-pool**	Displays information about the utilization of the MPLS label pool.

- Check the LSP connectivity.

Command	Function
Ruijie# **ping mpls ipv4** *ip-address/mask* [**repeat** *repeat*] [**ttl** *time-to-live*] [**timeout** *timeout*] [**size** *size*] [**interval** *mseconds*] [**source** *ip-address*] [**destination** *ip-address*] [**force-explicit-null**] [**pad** *pattern*] [**reply mode** { **ipv4** | **router-alert** }] [**dsmap**] [**flags fec**] [**verbose**]	Checks the LSP connectivity.
Ruijie# **traceroute mpls ipv4** *ip-address/mask* [**timeout** *timeout*] [**ttl** *ttl*] [**source** *ip-address*] [**destination** *ip-address*] [**force-explicit-null**] [**reply mode** { **ipv4 |** **router-alert** }] [**flags fec**] [**verbose**]	Checks the LSR nodes that the LSP passes through.

Configuring Optional LDP Parameters (Optional)

You can modify the default LDP parameter settings as required. To modify LDP parameters, use the commands in LDP configuration or interface configuration mode.

Configuring Parameters for an LDP Session

Configuring the LDP Router ID

The LDP router ID, expressed in the format of IP addresses, uniquely identifies one LSR in a domain. By default, the system router ID is used as the LDP router ID, that is, the LSR ID. The value of an LDP router ID must be globally unique. In addition, the LDP router ID must be reachable to other LSRs because LDP uses the LDP router ID as the transport address by default. To modify the LSR ID, use the **ldp router-id** command.

Command	Function
Ruijie(config-mpls-router)# **ldp router-id interface** *interface-name* [**force**]	Specifies the address of an interface as the LDP router ID of the LSR. **force** indicates that the current configurations immediately take effect.
Ruijie(config-mpls-router)# **no ldp router-id**	Restores the default value. The system router ID is used as the LDP router ID.

Configuring transport-address

By default, the LSR ID is used as the global transport address. As an option, you can choose the main address of an interface or specify an IP address as the transport address to set up an LDP session on the interface. The following shows two configuration methods.

Use the following commands to configure the transport address of an interface.

Command	Function
Ruijie(config-if-*type ID*)# **mpls ldp transport-address** { **interface** | *ip-address*}	Configures the transport address for LDP sessions on an interface.
Ruijie(config-if-*type ID*)# **no mpls ldp transport-address**	Deletes the configuration on the interface. By default, the global transport address is used. If no global transport address is configured, the LSR ID is used as the transport address.

Use the following commands to globally configure a transport address for all LDP sessions in LDP configuration mode.

Command	Function
Ruijie(config-mpls-router)# **transport-address** { **interface** | *ip-address* | *interface-name* }	Configures a global transport address for LDP sessions.
Ruijie(config-mpls-router)# **no transport-address**	Removes the global setting and restores the LSR ID as the transport address.

Caution

1. When you specify an IP address as the transport address, ensure that the address is reachable to other directly connected LSRs; otherwise, the LDP session cannot be set up.
2. If transport addresses are configured on an interface and globally, the basic LDP session set up on the interface prefers the transport address configured for the interface.
3. The configured transport address is valid only for the basic LDP session. The LDP session set up through extended mechanisms use the LSR ID as the transport address.

Configuring the Time Interval for Hello Packets

LDP periodically sends Hello packets to detect LDP peers. By default, the interval for sending Hello packets in the basic LDP discovery mechanism is 5 seconds. You can freely set the interval that ranges from 1 to 65535 seconds in interface mode.

Command	Function
Ruijie(config-if-type ID)# mpls ldp hello-interval seconds	Sets the interval for sending Hello packets in the basic LDP discovery mechanism.
Ruijie(config-if-type ID)# no mpls ldp hello–interval	Restores the default interval for sending Hello packets in the basic LDP discovery mechanism.

The default interval for sending Hello packets in the extended LDP discovery mechanism is 10 seconds. To modify the interval, use the **discovery target-hello** *interval* command.

Command	Function
Ruijie(config-mpls-router)# **discovery target-hello interval** *seconds*	Sets the interval for sending Hello packets in the extended LDP discovery mechanism.
Ruijie(config-mpls-router)# **no discovery target-hello interval**	Restores the default interval for sending Hello packets in the extended LDP discovery mechanism.

Configuring the Hold Time of Hello Packets

After an LDP peer is detected by periodically sent Hello packets, the local LDP device retains the peer for a period of time although no Hello packet is received from the peer, and considers that the peer expires after this period. This period of time is called the hold time of Hello packets. The default hold time of Hello packets is 15 seconds. You can freely set the interval that ranges from 1 to 65535 seconds in interface mode. The value 65535 indicates an indefinite hold time.

Command	**Function**
Ruijie(config-if-*type ID*)# **mpls ldp hello-holdtime** *seconds*	Sets the hold time of Hello packets.
Ruijie(config-if-*type ID*)# **no mpls ldp hello-holdtime**	Restores the default hold time of Hello packets.

The default hold time of Hello packets in the extended LDP discovery mechanism is 45 seconds. To modify this value, use the **discovery target-hello** *holdtime* command.

Command	**Function**
Ruijie(config-mpls-router)# **discovery target-hello holdtime** *seconds*	Sets the hold time of Hello packets in the extended LDP discovery mechanism.
Ruijie(config-mpls-router)# **no discovery target-hello holdtime**	Restores the default hold time of Hello packets in the extended LDP discovery mechanism.

Configuring the Hold Time of Keepalive Packets

After an LDP peer is detected by periodically sent Hello packets and an LDP session is set up in TCP mode, the local LDP device retains the peer for a period of time although no Keepalive packet is received from the peer. The local LDP device considers that the peer expires and voluntarily terminates the LDP session after this period. This period of time is called the hold time of Keepalive packets. The default hold time of Keepalive packets for the session set up in the basic discovery mechanism is 45 seconds and that for the session set up in the extended discovery mechanism is 180 seconds. You can freely set the value at the range of 15 to 65535. The interval for sending Keepalive packets is one third of the hold time of Keepalive packets.

Command	Function
Ruijie(config-if-*type ID*)# **mpls ldp keepalive–holdtime** *seconds*	In interface mode, sets the hold time of Keepalive packets for the session set up in the basic discovery mechanism.
Ruijie(config-if-*type ID*)# **no mpls ldp keepalive–holdtime**	Restores the default hold time of Keepalive packets for the session set up in the basic discovery mechanism.
Ruijie(config-mpls-router)# **targeted-session holdtime** *seconds*	In LDP mode, sets the hold time of Keepalive packets for the session set up in the extended discovery mechanism.
Ruijie(config-mpls-router)# **no targeted-session holdtime**	Restores the default hold time of Keepalive packets for the session set up in the extended discovery mechanism.

Configuring the Maximum Number of Repeated Label Requests

When an LDP device requests labels, it waits for a period of time to start another attempt if no label is detected due to various reasons. The default number of repeated requests is indefinite. You can freely set the value that ranges from 0 to 255 in interface mode.

Command	**Function**
Ruijie(config-mpls-router)# **mpls ldp max-label-requests** *times*	Sets the maximum number of repeated LDP label requests.

Ruijie(config-mpls-router)# **no mpls ldp max-label-requests**	Restores the default number of repeated LDP label requests.

Configuring the Maximum PDU

The messages exchanged between LDP devices are all contained in PDUs. You can freely set the value of the PDU that ranges from 256 to 4096 in interface mode. The default PDU value is 4096.

Command	Function
Ruijie(config-mpls-router)# **mpls ldp max-pdu** *max-pdu*	Sets the maximum PDU.
Ruijie(config-mpls-router)# **no mpls ldp max-pdu**	Restores the default PDU (4096).

Configuring the Extended LDP Discovery Mechanism

The basic discovery mechanism is used to detect the local LDP peers. That is, set up a local LDP session with the directly connected LSR. The extended discovery mechanism is used to detect the remote LDP peers. That is, set up a remote LDP session with the non-directly connected LSR.

Command	Function
Ruijie(config-mpls-router)# **neighbor** *ip-address*	Creates an extended LDP peer.
Ruijie(config-mpls-router)# **no neighbor** *ip-address*	Deletes an extended LDP peer.

Configuring LDP Loop Detection

Configuring the Loop Detection Mode

LDP provides two methods to detect loops: maximum number of hops and path vector. By default, loop detection is disabled for LDP.

In the loop detection based on the maximum number of hops, a packet carries both label information and the number of hops. The number increases by one every time the packet passes an LSR. When the number exceeds a preconfigured maximum value, the device considers that a loop occurs on the LSP.

In the loop detection based on the path vector, the packet carries the LSR ID apart from the label information. At each hop, an LSR first checks whether the number of LSRs in the path vector list exceeds the preset maximum number in the path vector list. If yes, a loop occurs. If no, the LSR continues to check whether its LSR ID exists in the path vector list of the LDP message. If yes, a loop occurs; if no, the LSR adds its own LSR ID to the path vector list.

Command	Function
Ruijie(config-mpls-router)# **loop-detection**	Enables loop detection.
Ruijie(config-mpls-router)# **no loop-detection**	Disables loop detection.

Configuring the Maximum Number of Hops

In interface mode, you can set the maximum number of hops allowed in loop detection mode. By default, the number is 254. You can set the value at the range of 1 to 255. If loop detection is enabled and the number of hops in an LDP message is detected to exceed the preset value, the LSR considers that a loop occurs.

Command	Function
Ruijie(config-if-*type ID*)# **mpls ldp max-hop-count** *number*	Sets the maximum number of hops in loop detection.

Ruijie(config-if-*type ID*)# **no mpls ldp max-hop-count**	Restores the default value of the maximum number of hops in loop detection.

Configuring the Maximum Number in the Path Vector List

In interface mode, you can set the maximum number of LSRs included in the path list of the loop detection based on path vector. By default, the number is 254. You can set the number at the range of 0 to 254. The number means the maximum number of LSRs that can be carried in the path vector list. After loop detection is enabled, an LSR considers that a loop occurs if the LSR detects its own LSR ID in the path vector list or the number of LSR IDs in the path vector list exceeds the preset value.

Command	Function
Ruijie(config-mpls-router)# **mpls ldp max-path-vector** *number*	Sets the maximum number in the path vector list of loop detection.
Ruijie(config-mpls-router)# **no mpls ldp max-path-vector**	Restores the default value of the maximum number in the path vector list of loop detection.

Configuring the LDP Working Mode

Configuring the LDP Label Distribution Control Mode

The LDP label distribution control mode specifies when an LSR notifies its neighbors of the binding between labels and FECs. There are two control modes: independent control and ordered control.

In independent control mode, the LSR can announce the binding of labels and FECs to its neighbors at any required time. In ordered control mode, an LSR binds an FEC to a label and sends the binding upstream only when the FEC has the next-hop label mapping or the LSR is the egress LSR of the FEC.

By default, LDP uses the independent control mode. To configure the LDP control mode, use the **lsp-control-mode** command.

Command	**Function**
Ruijie(config-mpls-router)# **lsp-control-mode** { **independent** | **orderd** }	Configures the label distribution control mode.
Ruijie(config-mpls-router)# **no lsp-control-mode**	Restores the default label distribution control mode.

Configuring the LDP Label Distribution Mode

The LDP label distribution mode specifies how an LSR notifies its neighbors of the binding between labels and FECs. There are two modes: DOD and DU.

In DOD mode, a downstream LSR responds to a label binding message only after receiving a label request from an upstream LSR. In DU mode, one LSR voluntarily sends label binding messages to its upstream LSRs according to certain triggering policies. If the upstream and downstream LSRs use different label distribution modes, the DU mode is used if the LSRs are connected to each other through Ethernet.

By default, LDP works in DU mode. You can use the **mpls ldp distribution-mode** command in interface mode to set the label distribution mode on an interface.

Command	**Function**
Ruijie(config-if-*type ID*)# **mpls ldp distribution-mode { du | dod }**	Configures the label distribution mode.

Ruijie(config-if-*type ID*)# **no mpls ldp distribution-mode**	Restores the default label distribution mode (DU).

Configuring the LDP Label Retention Mode

The label retention mode specifies whether an LSR should retain the label binding learnt from a label mapping message if the message is not sent from the next hop of the corresponding FEC or the message does not match any existing IP route. There are two label retention modes: conservative and liberal modes.

When the preceding situations occur, the liberal mode retains the binding of the FEC and label from the neighbor whereas the conservative mode does not retain the binding information.

The conservative label retention mode uses and maintains a small number of labels. The LSR should re-obtain the label values in case of route changes, prolonging responses. The liberal label retention mode, however, responds rapidly to route changes but unnecessary label mappings are also distributed and maintained.

By default, LDP uses the liberal label retention mode.

Use the **label-retention-mode** command to configure the label retention mode.

Command	Function
Ruijie(config-mpls-router)# **label-retention-mode** { **liberal | conservation** }	Configures the label retention mode.
Ruijie(config-mpls-router)# **no label-retention-mode**	Restores the default label retention mode.

Configuring Label Merging

If an LSR binds several incoming labels for a FEC but uses the same outgoing label for all packets in the FEC, it indicates that the LSR is capable of label merging. You can enable or disable label merging through LDP configurations.

By default, label merging is enabled for LDP.

Use the **label-merge** command to enable or disable label merging.

Command	Function
Ruijie(config-mpls-router)# **label-merge**	Enables label merging.
Ruijie(config-mpls-router)# **no label-merge**	Disables label merging.

Configuring the Transmission Mode of Label Release Messages

When an FEC becomes invalid, LDP sends label release messages to downstream devices to cancel the label bound to the FEC. Each LDP device on the LSR determines whether to transmit the messages to downstream devices based on the transmission mode of label release messages.

By default, an LDP device does not send label release messages received from an upstream device to downstream devices.

Use the **propagate-release** command to configure the transmission mode of label release messages.

Command	Function
Ruijie(config-mpls-router)# **propagate-release**	Configures a device to send label release messages to downstream devices.
Ruijie(config-mpls-router)# **no propagate-release**	Configures a device not to send label release messages to downstream devices.

Configuring Label Control Policies

Configuring Label Distribution Policies

By default, LDP assigns labels to all valid IGP routes (excluding BGP routes). In some special situations, you may only want to assign labels to some routes or to certain LDP peers to reduce the number of labels and the number of LSPs to lessen device and network burdens.

Command	Function
Ruijie(config-mpls-router)# **advertise-labels for host-routes**	Configures the device to assign labels to only host routes that satisfy the mask length of 32 bits in the route forwarding table. By default, the mask length of routes is not restricted.
Ruijie(config-mpls-router)# **advertise-labels for bgp-routes** [**acl** *acl-name*]	Configures the device to assign labels to BGP routes. By default, LDP does not assign labels to BGP routes.
Ruijie(config-mpls-router)# **advertise-labels for default-route**	Configures the device to assign labels to default routes. By default, LDP assigns implicit null label 3 to default routes.
Ruijie(config-mpls-router)# **advertise-labels for acl** *prefix-access-list* [**to** *peer-access-list*]	Configures the device to assign labels to FECs that match ACL rules and specify the device to assign labels only to LDP peers that match the rules.

Caution

1. By default, LDP assigns labels to only IGP routes. To assign labels to BGP routes, use the **advertise-labels for bgp-routes** command.
2. By default, LDP does not set up an LSP for default routes.

Configuring Label Reception Policies

By default, LDP can receive all label binding information sent from all neighbors. In certain situations, you may need to control the device to receive only some binding information about FECs and labels from certain neighbors. In this case, use the **neighbor** *ip-address* **labels accept** command.

Command	Function
Ruijie(config-mpls-router)# **neighbor** *ip-address* **labels accept** *acl-name*	Configures a label reception policy.

Configuring Policies for Distributing Explicit Null Labels

By default, LDP assigns implicit null labels to the FEC (such as direct routes) with the local device as the egress. You can use the **explicit-null** command to assign explicit null labels to all direct routes or routes that match certain ACL rules. To restore the default setting, use the **no explicit-null** command.

Command	Function
Ruijie(config-mpls-router)# **explicit-null** [**for** *prefix_acl* **| to** *peer_acl*]	Configures a device to assign explicit null labels to all direct routes or routes that match certain ACL rules.
Ruijie(config-mpls-router)# **no explicit-null**	Restores the default setting.

Caution

1. For an FEC with the local device as the egress, the device cannot assign explicit null labels to the FEC if the corresponding LSP is a tunnel that carries L2VPN or L3VPN services.
2. Configure this function only for the global LDP instance. This function is not supported by the VRF instance.

Configuring the LDP MD5 Authentication

To enhance the reliability of LDP sessions, you can configure the MD5 authentication for the TCP connections used by the LDP sessions. To configure the MD5 authentication for TCP connections between a device and its peer, use the **neighbor** *ip-address* **password** [**0 | 7**] *pwd-string* command. To restore the default setting, use the **no neighbor** *ip-address* **password** [**0 | 7**] *pwd-string* command.

Command	Function
Ruijie(config-mpls-router)# **neighbor** *ip-address* **password** [**0 | 7**] *pwd-string*	Configures a device to adopt the MD5 authentication for the TCP connections with its peer.
Ruijie(config-mpls-router)# **no neighbor** *ip-address* **password**	Restores the default setting.

Verifying the LDP Information

- Displaying LDP attributes

Use the **show mpls ldp parameters** [all **| vrf** *vrf-name*] command to view information about LDP attributes, including the LSR ID, transport address, loop detection mechanism, label distribution control mode, label retention mode, the interval and hold time of Hello packets with extended peers, and the interval and hold time of Keepalive packets with extended peers. You can verify the information to check whether the configurations are correct. By default, the LDP attributes of the default VRF are displayed. If **all** is chosen, the LDP attributes of all VRFs are displayed; if *vrf-name* is specified, the LDP attributes of a specified VRF are displayed.

Command	Function
Ruijie# **show mpls ldp parameters** [**all | vrf** *vrf-name*]	Displays information about LDP attributes.

- Displaying information about an interface enabled with LDP

Use the **show mpls ldp interface** [**all** | **vrf** *vrf-name* | *interface-name*] command to display the LDP status of interfaces in all or a specified VRF. You can also query the LDP status of specific interfaces. By default, the interface LDP status of interfaces in the default VRF is displayed. If **all** is chosen, the LDP status of interfaces in all VRFs is displayed; if *vrf-name* is specified, the LDP status of interfaces in a specified VRF is displayed; if *interface-name* is specified, the LDP status of the specified interface is displayed.

Command	Function
Ruijie# **show mpls ldp interface** [**all | vrf** *vrf-name* **|** *interface-name*]	Displays information about the interface enabled with LDP.

- Displaying the binding between FECs and labels

Use the **show mpls ldp bindings** [**all** |**vrf** *vrf-name*] | [*ip-address*/*mask* | **label** *label*] | [**remote** | **local**] command to display the binding information between FECs and labels. You can also use this command to view the LDP working status, to check whether an FEC is properly bound to a label, or query the specific label value bound to an FEC. When using this command, you can filter the display information based on the VRF, address prefix, label value, remote binding, or local binding.

Command	Function
Ruijie# **show mpls ldp bindings** [**all | vrf** *vrf-name*] | [*ip-address/mask* | **label** *label* **|** [**remote | local**]	Displays the binding between FECs and labels.

- Displaying LDP neighbors

Use the **show mpls ldp neighbor** [**all** \| **vrf** *vrf-name*] [*ip-address*] [**detail**] command to view the LDP neighbors of all or a specified VRF, including the TCP connection port, LDP status, statistics about packets received and transmitted, and the voluntary LDP discovery party of the local and remote LDP devices. If parameter **detail** is added, detailed information about LDP neighbors is displayed.

Command	Function
Ruijie# **show mpls ldp neighbor** [**all | vrf** *vrf-name*] | [**detail**]	Displays information about LDP neighbors.

- Displaying information about discovered LDP neighbors

Use the **show mpls ldp discovery** [**all** \| **vrf** *vrf-name*] \| [**detail**] command to display ports where LDP neighbors are discovered and information about the neighbors. If parameter **detail** is added, detailed information about LDP neighbors is displayed.

Command	Function
Ruijie# **show mpls ldp discovery** [**all | vrf** *vrf-name*] **|** [**detail**]	Displays information about discovered LDP neighbors.

- Resetting the LDP session

Use the **clear mpls ldp neighbor** command to reset an LDP session and set up a new session.

Command	Function
Ruijie# **clear mpls ldp neighbor** [**all** | **vrf** *vrf-name*] [*| *ip-address*]	Resets an LDP session and sets up a new session.

Configuring Static MPLS Forwarding

To support basic MPLS forwarding functions, you can also use static configurations other than LDP. To configure basic MPLS forwarding functions in static mode, perform the following configuration procedures:

- (Mandatory) Enable MPLS forwarding globally.
- (Mandatory) Enable MPLS forwarding on an interface.
- (Mandatory) Configure a static LSP.

Caution

LDP is not required when configuring a static LSP. As a result, IPv4 routes are also not required. If no IPv4 route exists on the network, the static LSP can still take effect as long as the physical network is reachable.

For configuration procedures to enable MPLS forwarding globally or on an interface, refer to the basic procedures for configuring MPLS forwarding.

Configuring a Static LSP

The configuration of an MPLS network in static mode centers around the static LSP. The other configurations are the same as those of LDP. To configure a static LSP, perform the following steps:

- Configure a static FTN on the ingress LSR.
- Configure a static ILM on the intermediate LSR.
- Configure a static ILM on the PHP LSR.

Note

Label values 16 to 1024 are reserved for static LSPs. You can select only from these label values when configuring static LSPs.

☑ On a router, use the **ip ref** command on the label forwarding interface to enable MPLS fast forwarding to improve forwarding performance of the router.

Configuring a Static FTN on the Ingress LSR

On the ingress LSP, set up an FTN entry for the FEC, that is, bind the FEC to a label.

Use the **mpls static ftn** command to configure a static FTN in global configuration mode. The syntax of the command is as follows:

Command	Function
Ruijie(config)# **mpls static ftn** *ip-address / mask* **out-label** *label* **nexthop** *interface nexthop-ip*	Adds a global FTN.
Ruijie(config)# **no mpls static ftn** *ip-address/mask*	Deletes a global FTN.

For example, to configure a global FTN that binds label 16 to FEC 192.168.1.0/24 and supports the next hop of the LSP as 192.168.10.10 and the outgoing interface as GigabitEthernet 2/1, run the following command:

```
Ruijie(config)# mpls static ftn 192.168.1.0/24 out-label 16 GigabitEthernet 2/1 192.168.10.10
```

To delete the TFN, run the following command. In this case, you are required to only enter the FEC. Other parameters are not required.

```
Ruijie(config)# no mpls static ftn 192.168.1.0/24
```

Configuring a Static ILM on the Intermediate LSR

An intermediate LSR forwards labels for incoming labeled packets. In this case, you are required to configure ILM forwarding entries to map incoming labels to outgoing ones. Use the **mpls static ilm in-label** command to configure a static ILM in global configuration mode. The syntax of the command is as follows:

Command	Function
Ruijie(config)# **mpls static ilm in-label** *in_label* **forward-action swap-label** *swap_label* **nexthop** *interface nexthop-ip* **fec** *ip-address/mask*	Adds a global ILM.
Ruijie(config)# **no mpls static ilm in-label** *in_label*	Deletes a global ILM.

For example, to configure a global ILM that maps the incoming label 16 to the outgoing label 17 and supports the next hop of the LSP as 192.168.11.11, the outgoing interface as GigabitEthernet 2/2, and the FEC of the LSP as 192.168.1.0/24, run the following command:

```
Ruijie(config)# mpls static ilm in-label 16 forward-action swap-label 17 nexthop
GigabitEthernet 2/2 192.168.11.11 fec 192.168.1.0/24
```

To delete the ILM, run the following command:

```
Ruijie(config)# no mpls static ilm in-label 16
```

Configuring a Static ILM on the PHP LSR

Because the second but last hop is required to perform PHP, its ILM entries must be different from those on other intermediate LSRs. That is, the outgoing label in the ILM of the PHP LSR on the LSP should be an implicit null label (3).

Note

For information about the PHP, refer to related materials.

For example, you are required to configure a global ILM on the PHP LSR of the LSP. The LSR is required to pop out the incoming label 17 and send the packets from GigabitEthernet 2/2. The next hop address is 192.168.12.12 and the corresponding FEC is 192.168.1.0/24. Run the following command:

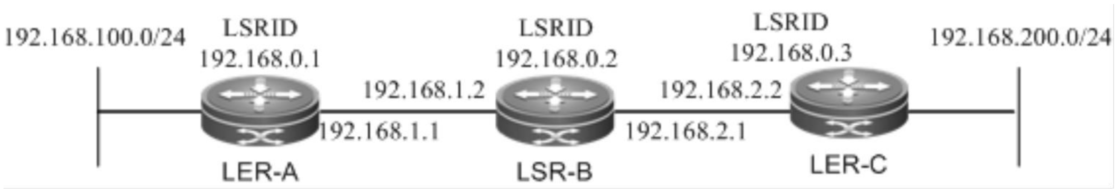
```
Ruijie(config)# mpls static ilm in-label 17 forward-action swap-label 3 nexthop GigabitEthernet
2/2 192.168.11.11 fec 192.168.1.0/24
```

To delete the ILM, run the following command:

```
Ruijie(config)# no mpls static ilm in-label 17
```

Example for Configuring Basic MPLS Functions

Figure 4



As shown in the preceding figure, three MPLS devices are deployed to construct an MPLS network. The following section introduces the setup of an LSP through LDP and the setup of a static LSP to show the MPLS configuration procedures.

Setting Up an LSP Through LDP

LDP works only with IPv4 routes. Here, OSPF is enabled to set up IPv4 routes. Before the following configuration, ensure that you have created a loopback interface (Loopback 0) and assigned an IP address, which also serves as the router ID, to the loopback interface on each device.

Configurations on LER_A:

Command	Function

Command	Function
Ruijie(config)# **mpls ip**	Enables MPLS forwarding globally. (Note: This command is not applicable to switch chip forwarding.)
Ruijie(config)# **mpls router ldp**	Enables LDP and enters LDP mode.
Ruijie(config-mpls-router)# **ldp router-id interface loopback** 0 **force**	Configures the LDP router ID. The loopback address is generally used as the router ID.
Ruijie (config-mpls-router)# **exit**	Exits LDP mode and enters global configuration mode.
Ruijie(config)# **interface gigabitEthernet** *2/2*	Enters interface mode of GigabitEthernet 2/2.
Ruijie(config-if-GigabitEthernet 2/2)# **mpls ip**	Enables LDP on the interface.
Ruijie(config-if-GigabitEthernet 2/2)# **label-switching**	Enables MPLS on the interface.
Ruijie(config-if-GigabitEthernet 2/2)# **ip ref**	Enables MPLS fast forwarding on the interface of a router.
Ruijie(config-if-GigabitEthernet 2/2)# **exit**	Exits interface mode and enters global configuration mode.
Ruijie (config)# **router ospf** *10*	Enables OSPF and enters OSPF mode.
Ruijie (config-router)# **network** *192.168.100.0 0.0.0.255* **area 0** Ruijie (config-router)# **network** *192.168.0.1 0.0.0.0* **area 0** Ruijie (config-router)# **network** *192.168.1.0 0.0.0.255* **area 0**	Adds routing information to OSPF.
Ruijie(config-router)# **end**	Ends the configuration.

Configurations on LER_B:

Command	Function
Ruijie (config)# **mpls ip**	Enables MPLS forwarding globally. (Note: This command is not applicable to switch chip forwarding.)
Ruijie (config)# **mpls router ldp**	Enables LDP and enters LDP mode.
Ruijie(config-mpls-router)# **ldp router-id interface loopback** *0* **force**	Configures the LDP router ID. The loopback address is generally used as the router ID.
Ruijie (config-mpls-router)# **exit**	Exits LDP mode and enters global configuration mode.
Ruijie (config)# **interface gigabitEthernet** *2/1*	Enters interface mode of GigabitEthernet 2/1.
Ruijie(config-if-GigabitEthernet 2/1)# **mpls ip**	Enables LDP on the interface.
Ruijie(config-if-GigabitEthernet 2/1)# **label-switching**	Enables MPLS on the interface at the public network side.
Ruijie(config-if-GigabitEthernet 2/2)# **ip ref**	Enables MPLS fast forwarding on the interface of a router.
Ruijie(config-if-GigabitEthernet 2/1)# **exit**	Exits interface mode and enters global configuration mode.
Ruijie (config)# **router ospf** *10*	Enables OSPF and enters OSPF mode.
Ruijie (config-router)# **network** *192.168.1.0 0.0.0.255* **area** *0* Ruijie (config-router)# **network** *192.168.2 .0 0.0.0.255* **area** *0* Ruijie (config-router)# **network** *192.168.0.2 0.0.0.0.0* **area** *0*	Adds routing information to OSPF.
Ruijie (config-router)# **end**	Ends the configuration.

Configurations on LER_C:

Command	Function
Ruijie (config)# **mpls ip**	Enables MPLS forwarding globally. (Note: This command is not applicable to switch chip forwarding.)
Ruijie (config)# **mpls router ldp**	Enables LDP and enters LDP mode.
Ruijie(config-mpls-router)# **ldp router-id interface loopback** *0* **force**	Configures the LDP router ID. The loopback address is generally used as the router ID.
Ruijie (config-mpls-router)# **exit**	Exits LDP mode and enters global configuration mode.
Ruijie (config)# **interface gigabitEthernet** *2/1*	Enters interface mode of GigabitEthernet 2/1.
Ruijie(config-if-GigabitEthernet 2/1)# **mpls ip**	Enables LDP on the interface.
Ruijie(config-if-GigabitEthernet 2/1)# **label-switching**	Enables MPLS on the interface at the public network side.
Ruijie(config-if-GigabitEthernet 2/2)# **ip ref**	Enables MPLS fast forwarding on the interface of a router.
Ruijie(config-if-GigabitEthernet 2/1)# **exit**	Exits interface mode and enters global configuration mode.
Ruijie (config)# **router ospf** *10*	Enables OSPF and enters OSPF mode.
Ruijie (config-router)# **network** *192.168.200.0 0.0.0.255* area *0* Ruijie (config-router)**#** **network** *192.168.0.3 0.0.0.0* **area** *0* Ruijie (config-router)# **network** *192.168.2.0 0.0.0.255* **area** *0*	Adds routing information to OSPF.
Ruijie (config-router)# **end**	Ends the configuration.

Configuring a Static LSP

You can configure a static LSP without IPv4 routes.

For example, set up two LSPs between interface 1 at the 192.168.100.0/24 network segment on LER_A and interface 2 at the 192.168.200.0/24 network segment on LER_C to connect the two network segments. You need to configure one LSP from LER_A to LER_C and the other LSP from LER_C to LER_A. This is because the LSP is uni-directional.

☑ On a router, you must use the **ip ref** command on the label forwarding interface to enable MPLS fast forwarding to improve forwarding performance of the router.

Configurations on LER_A:

Command	Function
Ruijie (config)# **mpls ip**	Enables MPLS forwarding globally. (Note: This command is not applicable to switch chip forwarding.)
Ruijie (config)# **interface gigabitEthernet** *2/2*	Enters interface mode of GigabitEthernet 2/2.
Ruijie(config-if-GigabitEthernet 2/2)# **label-switching**	Enables MPLS on the interface at the public network side.
Ruijie(config-if-GigabitEthernet 2/2)# **ip ref**	Enables MPLS fast forwarding on the interface of a router.
Ruijie(config-if-GigabitEthernet 2/2)# **exit**	Exits interface mode and enters global configuration mode.

Command	Function
Ruijie (config)# **mpls static ftn** *192.168.200.0/24* **out-label** *16* **nexthop gigabitEthernet** *2/2 192.168.1.2*	Creates an FTN that binds 192.168.200.0/24 to label 16. Specifies the next hop of the FTN as 192.168.1.2 and the outgoing interface as GigabitEthernet 2/2.
Ruijie(config-router)# **end**	Ends the configuration.

Configurations on LER_B:

Command	Function
Ruijie (config)# **mpls ip**	Enables MPLS forwarding globally. (Note: This command is not applicable to switch chip forwarding.)
Ruijie (config)# **interface gigabitEthernet** *2/1*	Enters interface mode of GigabitEthernet 2/1.
Ruijie(config-if-GigabitEthernet 2/1)# **label-switching**	Enables MPLS on the interface at the public network side.
Ruijie(config-if-GigabitEthernet 2/1)**# ip ref**	Enables MPLS fast forwarding on the interface of a router.
Ruijie(config-if-GigabitEthernet 2/1)# **exit**	Exits interface mode and enters global configuration mode.
Ruijie (config)# **interface gigabitEthernet** *2/2*	Enters interface mode of GigabitEthernet 2/2.
Ruijie(config-if-GigabitEthernet 2/2)# **label-switching**	Enables MPLS on the interface at the public network side.
Ruijie(config-if-GigabitEthernet 2/1)**# ip ref**	Enables MPLS fast forwarding on the interface of a router.
Ruijie(config-if-GigabitEthernet 2/2)# **exit**	Exits interface mode and enters global configuration mode.
Ruijie (config)# **mpls static ilm in-label** *16* **forward-action swap-label** *3* **nexthop gigabitEthernet** *2/2 192.168.2.2* **fec** *192.168.200.0/24*	Creates an ILM that maps the incoming label 16 to the outgoing label 3 (implicit null label) on GigabitEthernet 2/2. Specifies the next hop address as 192.168.2.2 and the FEC as 192.168.200.0/24.
Ruijie (config)# **mpls static ilm in-label** *17* **forward-action swap-label** *3* **nexthop gigabitEthernet** *2/1 192.168.1.1* **fec** *192.168.100.0/24*	Creates an ILM that maps the incoming label 17 to the outgoing label 3 (implicit null label) on GigabitEthernet 2/1. Specifies the next hop address as 192.168.1.1 and the FEC as 192.168.100.0/24.
Ruijie (config-router)# **end**	Ends the configuration.

Because LER_B is the PHP LSR for the FEC 192.168.100.0/24, the incoming label 17 is mapped to the outgoing label 3 (implicit null label). The outgoing interface is GigabitEthernet 2/1.

Similarly, because LER_B is the PHP LSR for the FEC 192.168.200.0/24, the incoming label 16 is also mapped to the outgoing label 3 (implicit null label). The outgoing interface is GigabitEthernet 2/2.

Configurations on LER_C:

Command	Function
Ruijie (config)# **interface gigabitEthernet** *2/1*	Enters interface mode of GigabitEthernet 2/1.
Ruijie(config-if-GigabitEthernet 2/1)# **label-switching**	Enables MPLS on the interface at the public network side.
Ruijie(config-if-GigabitEthernet 2/1)# **ip ref**	Enables MPLS fast forwarding on the interface of a router.
Ruijie(config-if-GigabitEthernet 2/1)# **exit**	Exits interface mode and enters global configuration mode.
Ruijie (config)# **mpls static ftn** *192.168.100.0/24* **out-label** *17* **nexthop gigabitEthernet** *2/1 192.168.2.1*	Creates an FTN that binds 192.168.200.0/24 to label 17. Specifies the next hop of the FTN as 192.168.2.1 and the outgoing interface as GigabitEthernet 2/1.
Ruijie (config-router)# **end**	Ends the configuration.

After the preceding configurations, packets destined for the 192.168.200.0/24 network segment on LER_A are sent out through GigabitEthernet 2/2 on LER_A and pushed with label 16. After arriving at the GigabitEthernet 2/1 interface on LER_B, these packets with label 16 are then transformed to IP packets and sent out through GigabitEthernet 2/2 on LER_B. After these IP packets destined for the 192.168.200.0/24 network segment arrives at LER_C, LER_C selects routes based on the destination IP address and sends out the packets from GigabitEthernet 2/1.

Configuring **BGP/MPLS IP VPN**

Understanding the BGP/MPLS IP VPN

Overview

In traditional VPNs, private network data streams are generally transmitted over public networks through Generic Routing Encapsulation (GRE), Layer 2 Tunneling Protocol (L2TP), and Point-to-Point Tunneling Protocol (PPTE). As another implementation of a VPN, a BGP/MPLS IP VPN can be considered as a VPN between Layer 2 and Layer 3. A label switched path (LSP) is a tunnel that is set up through MPLS LDP on the public network. In an MPLS VPN, the different branches of private networks at different locations are connected to form one network through LSPs. The MPLS VPN also supports interworking between different VPNs. The implementation of the VPN through MPLS has natural advantages. For VPN users, the workload is largely reduced because no special VPN devices are required to construct the VPN. Instead, the VPN users can directly use traditional routers. For carriers, the MPLS VPN can be easily expanded.

As a highly effective technical platform for IP backbone networks, MPLS provides VPNs with flexible and scalable technical foundations.

The L3VPN based on BGP/MPLS IP VPN has the following features:

- The VPN tunnels are set up on the provider edge (PE) devices of network service providers rather than the customer edge (CE) devices. The VPN routes are also transmitted between PEs. In this manner, users are not required to maintain VPN information.
- Existing routing protocols are directly utilized. The setup of VPN tunnels and route advertising are dynamically implemented, facilitating the expansion of VPNs.
- Address overlapping is supported. Different VPN users can use the same address space.
- On the network of service providers, VPN services are exchanged by label switching rather than traditional route distribution.
- The L3VPN is as secure as user dedicated lines.

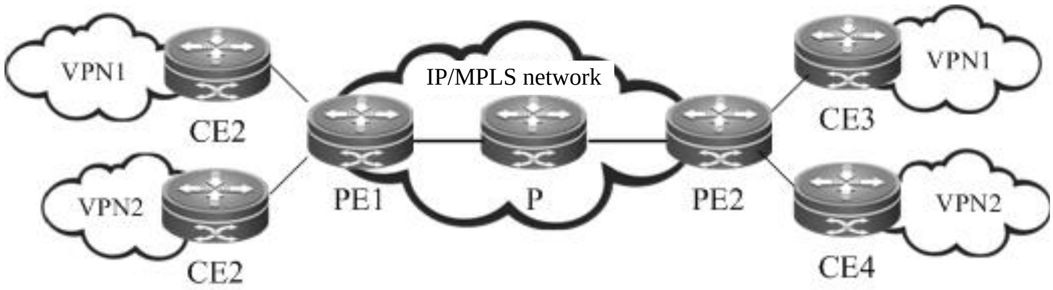
The BGP/MPLS IP VPN provides the following functions:

- Adopts the LDP to set up LSPs on the backbone network. This process is generally performed on the provider's network and completed when the topology becomes stable.
- Forwards data packets based on the pushed labels and the local mapping table.
- Supports MP-BGP and extended BGP attributes to transmit VPN routes and carry VPN attributes and labels.
- Manages VPN routes to set up multiple routing tables and maintain VPN routes.

Components of the BGP/MPLS IP VPN

A BGP/MPLS IP VPN model consists of three components, as shown in the following figure.

Figure 5 Basic components of the VPN



- CE

Located at a customer edge, a CE logically belongs to a user VPN. One interface on the CE is directly connected to a PE device. The CE can be a host, router or switch that may not support MPLS. As shown in the figure, CE1, CE2, CE3, and CE4 are CE devices.

- PE

A PE is an edge device on the SP backbone network. It can be a router, an ATM switch, or an FR switch, as shown by PE1 and PE2 in the figure. A PE logically belongs to the service provider and is directly connected to a CE. You can connect one PE to multiple CEs. The PE is mainly responsible for receiving the VPN information from CEs and transmitting the information to other PEs, or receiving the VPN information from other PEs and distributing it to the CEs. The PEs should support MPLS.

- P

The provider router (P) is a core device on the SP backbone network, as shown by P1, P2, and P3 in the figure. The P is not connected to CEs. It is responsible for routing and rapid forwarding. As a device on the core MPLS backbone network, the P should support MPLS. The P knows the routes to any destination on the backbone network but does not know the routes to a VPN.

VRF

- VRF

The VPN routing and forwarding table (VRF) is used to solve the conflicts of local routes. Each connection between a PE and a CE is associated with a VRF. One PE can have several VRFs to exchange route information with CEs. You can consider a VRF as a virtual router. Each virtual router should be connected to a CE to receive route information from the CE or advertise the VPN route information to the CE. The VRF solves the conflicts of local routes due to the adoption of the same address space by different VPNs. One VRF includes:

5) An independent routing table

6) A group of interfaces that belong to the VRF

7) A group of routing protocols that are used in the VRF

The VRF has two important attributes: route distinguisher (RD) and route-target (RT) attributes.

- RD

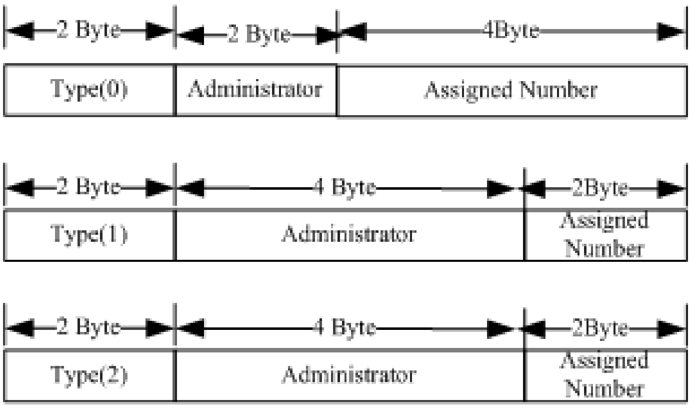
The RD is introduced to solve the conflict of routes during the transmission.

You can consider the RD as a distinguisher. If different VPNs use the same network address and advertise their route information on the backbone network through BGP, BGP chooses and advertises only the best route from the overlapped addresses. As a result, some VPNs cannot obtain their route information. If the RD values are added to the overlapped addresses, BGP identifies the same network addresses based on the different RDs carried in the VPN information. In this manner, each VPN can obtain its own route information. The RD only serves as a distinguisher to distinguish the same network addresses. If address overlapping does not exist for different VPNs, you can configure no RD values.

Generally, a unique RD value is specified for one VPN. In this manner, different VPNs have different RDs, facilitating the transmission of route information on the backbone network. The RD value is generally defined as xx: xx, such as RD 1: 100, where 1 stands for the AS number of the backbone network and 100 is a number specified by the user. One VPN route can carry only one RD value.

The RD consists of three fields: type, administrator, and assigned number. Based on the value of the type field, the encoding formats are classified into the following three types:

Figure 6 RD structure.



8) When Type = 0, the administrator field has two bytes and is marked by the AS number that must be a public AS number. The assigned number has four bytes and is managed by the service provider.

9) When Type = 1, the administrator field has four bytes and uses an IPv4 address that must be a global IP address. The assigned number has two bytes and is managed by the service provider.

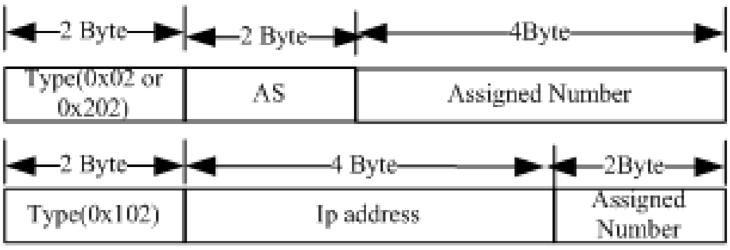
10) When Type = 2, the administrator field has four bytes and is marked by the four-byte AS number. The assigned number has two bytes and is managed by the service provider.

- Route-Target

The introduction of the RT attribute is to let the VRF choose its route selection mode. The RT attributes are classified into route-target export and route-target import. A PE receives routes from a CE and adds route-target export to the VPN routes and then advertises the VPN routes to other PEs. The PE determines whether to import the routes received from other PEs to the VRF based on the route-target import. One principle is that when a PE receives a VPN route, the PE imports the route to the VRF only when at least one RT attribute carried in the route is the same as the import RT in the VRF of the PE. In this manner, you can flexibly control the distribution of VPN routes. One VPN route can carry multiple RT values.

The BGP extended community attribute defines the RT encoding structure, as shown in the following figure.

Figure 7 RT structure



The definition of RT is similar to that of RD. For 0x02 and 0x202, the AS number must be a public one. For 0x102, the IPv4 address must be a global one rather than a private address.

MP-BGP

The VPN route information is transmitted on the backbone network through BGP. The export RT attribute is carried in the BGP extended community attribute. The traditional BGP4, however, transmits only IPv4 routes and cannot carry the VPN route that includes RDs. Therefore, the BGP is extended to introduce new attributes. One of the biggest advantages of BGP is its scalability. The Multi-Protocol (MP-BGP) is a new attribute introduced to the original BGP to support multiple protocols. The MP-BGP can carry VPN information. In this manner, the VPN route takes up the form of RD + IP address prefix. By adding RDs to VPN routes exchanged between PEs, the MP-BGP allows VPN users to change the IPv4 routes to VPN-IPv4 routes and transmit the routes on the backbone network.

Protocol Specification

- IETF RFC 4364: BGP/MPLS IP Virtual Private Networks (VPNs)

Configuring the BGP/MPLS IP VPN

☑ On a router, you must use the **ip ref** command on the interface to enable MPLS fast forwarding to improve forwarding performance of the router.

Default Configuration

BGP/MPLS L3 VPN is disabled by default.

Function	Default Setting
Basic BGP/MPLS L3 VPN functions	Disabled
VPN label distribution mode	Distributing labels to each VRF
Inter-AS VPN	Disabled
CSC VPN	Disabled
MPLS VPN over GRE	Disabled

Configuring the Basic BGP/MPLS IP VPN

To configure basic BGP/MPLS IP VPN functions, perform the following configurations:

- Configuring an MPLS Network (Mandatory)

- Configuring a VPN Routing Instance (Mandatory)
- Configuring PEs to Transmit VPN Routes (Mandatory)
- Configuring Route Exchange Between PEs and CEs (Mandatory)
- Configuring VPN Label Distribution Mode (Optional)
- Configuring Import and Export Policies for VPN Routes (Optional)
- No import VPN route-target community //Import extended community attribute list (not configured)
-

Configuring an MPLS Network

To use MPLS on the backbone network, you must configure the MPLS LDP on the P and PE to set up public tunnels. This means that you must configure LDP on MPLS devices and enable MPLS on each interface.

Use the following commands to configure an MPLS network.

Command	**Function**
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **mpls ip**	Enables MPLS globally. ⚠ Caution This command is not available on a switch chip.
Ruijie(config)# **mpls router ldp**	Enables LDP and enters LDP configuration mode.
Ruijie(config-mpls-router)# **ldp router-id interface loopback** *id* [**force**]	Configures the LDP router ID. The IP address of the loopback interface is generally used as the router ID.
Ruijie(config-mpls-router)# **exit**	Exits LDP configuration mode.
Ruiije(config)# **interface** *type ID*	Enters interface configuration mode.
Ruijie(config-if-*type ID*)# **ip address** *ip-address mask*	Assigns an IP address to the interface.
Ruijie(config-if-*type ID*)# **label-switching**	Enables MPLS on the interface at the public network side.
Ruijie(config-if-*type ID*)# **ip ref**	☑ In case of a router, enables MPLS fast forwarding on the interface.
Ruijie(config-if-*type ID*)# **mpls ip**	Enables LDP on the interface.
Ruijie(config-if-*type ID*)# **show running-config**	Displays existing configuration information.

# Configure an MPLS network.

```
Ruijie# configure terminal
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# exit
Ruijie(config)#interface gigabitethernet 1/1
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-gigabitethernet 1/1)# no switchport
Ruijie(config-if-gigabitethernet 1/1)# ip address 192.168.10.1 255.255.255.0
Ruijie(config-if-gigabitethernet 1/1)# label-switching
Ruijie(config-if-gigabitethernet 1/1)# mpls ip
```

Configuring a VPN Routing Instance

A VPN routing instance is the VRF that is configured on PEs. The CE and P devices do not have VRFs.

The configuration of a VRF includes defining the VRF, assigning RD and RT values to the VRF, and associating the VRF with an interface.

Use the following commands to configure a VPN routing instance.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **ip vrf** *vrf-name*	Creates a VRF and enters VRF configuration mode.
Ruijie(config-vrf)# **rd** *rd-value*	Sets the RD value.
Ruiije(config-vrf)# **route-target {both | export | import**} *rt-value*	Sets the RT value.
Ruijie(config-vrf)# **exit**	Exits VRF configuration mode.
Ruijie(config)# **interface type** *ID*	Enters interface configuration mode.
Ruijie(config-if-*type ID*)# **ip vrf forwarding** *vrf-name*	Associates the interface with the VRF.
Ruijie(config-if-*type ID*)# **ip address** *address mask*	Assigns an IP address to the interface.
Ruijie(config-if-*type ID*)# **ip ref**	☑ In case of a router, enables MPLS fast forwarding on the interface.
Ruijie(config-if-*type ID*)# **show running-config**	Displays existing configuration information.

# Configure a VRF and bind it to interface Gigabitethernet 1/1.

```
Ruijie(config)# ip vrf vpn1
Ruijie(config-vrf)# rd 100: 1
Ruijie(config-vrf)# route-target both 100: 1
Ruijie(config-vrf)# exit
Ruijie(config)# interface gigabitethernet 1/1
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-gigabitethernet 1/1)# no switchport
Ruijie(config-if-gigabitethernet 1/1)# ip vrf forwarding vpn1
Ruijie(config-if-gigabitethernet 1/1)# ip address 192.168.10.1
255.255.255.0
```

Caution

If an RD value is specified for the VRF on a PE or the BGP VRF function is enabled on the PE, the RD value cannot be modified or deleted. In this case, you can only delete the VRF and create the VRF again to set the RD value.

Two different VRFs on the same PE cannot be assigned the same RD.

If you enter the **ip vrf forwarding** *vrf-name* command, the IP address assigned to the interface earlier is deleted. In this case, you need to redefine the IP address in interface configuration mode.

Configuring PEs to Transmit VPN Routes

PEs transmit route information to each other through BGP. Because a PE needs to transmit VPN route information rather than common IPv4 route information to another PE, you need to enter VPN address family configuration mode to configure the PE to transmit VPN routes to the peer PE.

Use the following commands to configure the PE to transmit VPN routes to another PE.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **router bgp** *asn-num*	Creates a BGP domain and enters BGP configuration mode.
Ruijie(config-router)# **neighbor** *ip-address* **remote-as** *asn-number*	Configures a BGP session.
Ruiije(config-router)# **neighbor** *ip-address* **update-source** *interface-name*	Sets the interface address used to set up the MP-IBGP session as the source address. The address of the loopback interface is generally used as the source address.
Ruiije(config-router)# **address-family vpnv4**	Enters the VPN address family.
Ruijie(config-router-af)# **neighbor** *ip-address* **activate**	Activates the BGP session to exchange VPN routes.
Ruijie(config-router-af)# **show running-config**	Displays existing configuration information.

# Set up an MP-BGP session with the neighboring PE at 1.1.1.1.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 1.1.1.1 remote-as 1
Ruijie(config-router)# neighbor 1.1.1.1 update-source loopback 0
Ruijie(config-router)# address-family vpnv4
Ruijie(config-router-af)# neighbor 1.1.1.1 activate
```

Configuring Route Exchange Between PEs and CEs

Running BGP Between PEs and CEs to Transmit Route Information

To configure a BGP session with a CE, you need to enter VRF address family configuration mode on the PE and then configure the routing protocol with the CE.

Use the following commands on the PE to configure BGP.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **router bgp** *pe-asn-num*	Creates a BGP domain and enters BGP configuration mode.
Ruijie(config-router)# **address-family ipv4 vrf** *vrf-name*	Configures and enters BGP VRF address family configuration mode.
Ruiije(config-router-af)# **neighbor** *ip-address* **remote-as** *ce-asn-num*	Sets up an EBGP session with a CE.
Ruijie(config-router-af)# **show running-config**	Displays existing configuration information.

# Set up an EBGP session with the neighboring CE at 192.168.10.2.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# address-family ipv4 vrf vrf1
Ruijie(config-router)# neighbor 192.168.10.2 remote-as 2
```

Caution

If no RD is specified for the VRF, you will be reminded that no RD value is configured when you use the **address-family ipv4 vrf vrf-name** command to enter the specified VRF address family. As a result, you cannot enter the address family.

Use the following commands on the CE to configure a PE peer.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **router bgp** *ce-asn-num*	Creates a BGP domain and enters BGP configuration mode.
Ruiije(config-router)# **neighbor** *ip-address pe-asn*	Sets up an EBGP session with a PE.
Ruijie(config-router)# **show running-config**	Displays existing configuration information.

# Set up an EBGP session with the PE at 192.168.10.1.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 2
Ruijie(config-router)# neighbor 192.168.10.1 remote-as 1
```

Configuring OSPF Between PEs and CEs to Transmit Route Information

To run OSPF between a PE and a CE, you must configure an OSPF instance for the VRF on the PE. The VRF then uses the OSPF instance to exchange route information between the PE and the CE. By redistributing BGP routes, OSPF sends the VPN routes received from other PEs to the CE. At the same time, by redistributing OSPF routes, BGP sends the VPN route information that is distributed to the PE by the CE to other PE peers.

Use the following commands on the PE to configure OSPF.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **router ospf** *ospf-id* **vrf** *vrf-name*	Creates an OSPF instance and enters OSPF configuration mode.
Ruijie(config-router)# **network** *prefix mask* **area** *area-id*	Configures an OSPF link.
Ruiije(config-router)# **redistribute bgp subnets**	Configures OSPF to redistribute BGP routes.
Ruiije(config-router)# **exit**	Exits OSPF configuration mode.
Ruijie(config)# **router bgp** *asn*	Enables BGP and enters BGP configuration mode.
Ruijie(config-router)# **address-family ipv4 vrf** *vrf-name*	Enters BGP VRF configuration mode.
Ruijie(config-router-af)# **redistribute ospf** *ospf-id*	Redistributes OSPF routes.
Ruijie(config-router-af)# **show running-config**	Displays existing configuration information.

# Run OSPF between a PE and a CE to distribute VPN routes.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 10 vrf vrf1
Ruijie(config-router)# network 192.168.10.0 255.255.255.0 area 0
Ruijie(config-router)# redistribute bgp subnets
Ruijie(config-router)# exit
Ruijie(config)# router bgp 1
Ruijie(config-router)# address-family ipv4 vrf vrf1
Ruijie(config-router-af)# redistribute ospf 10
```

Configuring RIP Between PEs and CEs to Transmit Route Information

The PE and CE run RIP. The VRF on the PE can exchange route information between the PE and the CE. RIP redistributes BGP routes and transmits VPN routes from other PEs to the CE. Meanwhile, BGP redistributes RIP routes and transmits VPN routes distributed by the CE to the PE to other PE peers.

Use the following commands on the PE to configure RIP.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **router rip**	Creates a RIP instance and enters RIP configuration mode.
Ruiije(config-router)# **address-family ipv4 vrf** *vrf-name*	Enters RIP VRF address family configuration mode.
Ruijie(config-router)# **version 2**	Configures the version of RIP.
Ruiije(config-router-af)# **network** *network-number* [*wildcard*]	Configures RIP on the PE and CE.
Ruiije(config-router-af)# **redistribute bgp**	Configures RIP to redistribute BGP routes.
Ruiije(config-router-af)# **exit**	Exits address family configuration mode.
Ruijie(config)# **router bgp** *asn*	Configures BGP and enters BGP configuration mode.
Ruijie(config-router)# **address-family ipv4 vrf** *vrf-name*	Enters BGP VRF configuration mode.

Command	Function
Ruijie(config-router-af)# **redistribute rip**	Redistributes OSPF routes.
Ruijie(config-router)# **show running-config**	Displays existing configuration information.

# Run RIP between a PE and a CE to distribute VPN routes.

```
Ruijie# configure terminal
Ruijie(config)# router rip
Ruijie(config-router)# address-family ipv4 vrf vrf1
Ruijie(config-router-af)# version 2
Ruijie(config-router-af)# network 192.168.10.0
Ruijie(config-router-af)# redistribute bgp
Ruijie(config-router-af)# exit-address-family
Ruijie(config)# router bgp 1
Ruijie(config-router)# address-family ipv4 vrf vrf1
Ruijie(config-router-af)# redistribute rip
```

Transmitting Route Information Between a PE and a CE Through Static Configurations

In simple network environments, you can generally configure static routes. Use the following commands to configure a static route.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **ip route vrf** vrf-name prefix mask interface-name nexthop	Configures a static route.
Ruijie(config)# **router bgp** *asn*	Enters BGP configuration mode.
Ruijie(config-router)# **address-family ipv4 vrf** *vrf-name*	Enters BGP VRF address family configuration mode.
Ruijie(config-router-af)# **redistribute static**	Redistributes static routes.
Ruijie(config-router)# **show running-config**	Displays existing configuration information.

# Configure a static route on the PE to distribute VPN routes.

```
Ruijie# configure terminal
Ruijie(config)# ip router vrf vrf1 192.168.20.0 255.255.255.0 gigabitEthernet 2/3 192.168.10.2
Ruijie(config)# router bgp 1
Ruijie(config-router)# address-family ipv4 vrf vrf1
Ruijie(config-router-af)# redistribute static
```

Configuring VPN Label Distribution Mode (Optional)

RFC 4364 describes two label distribution modes for L3VPN applications: route-based label distribution and VRF-based label distribution. The advantage of the former is rapid forwarding that allows a device to forward packets to the next hop by searching the ILM table. The disadvantage, however, is the large capacity of the ILM table. The advantage of the latter is the reduced capacity of the ILM table. This is because one label is assigned to each VRF and all routes in the VRF thus share the label. The disadvantage is the lower forwarding efficiency because it requires two times of table searching. The device should first locate the VRF of the packets based on the ILM table and then forward the packets by searching routes in the VRF based on the destination IP address.

An L3VPN adopts VRF-based label distribution mode by default. You can run the **alloc-label** command in VRF configuration mode to change default label distribution mode. You can also choose different distribution modes for different VRFs.

To configure label distribution mode, enter privileged EXEC mode and use the following commands.

Command	Function
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **ip vrf** *vrf-name*	Creates a VRF and enters VRF configuration mode.
Ruijie(config-vrf)# alloc-label per-vrf	Assigns one label to all routes in the VRF. When distributing VPN routes, MP-BGP uses the same label for all routes.
Ruijie(config-vrf)# **alloc-label per-route**	Assigns one label to each route in the VRF. When distributing VPNv4 routes, MP-BGP uses a different label for each route.
Ruijie(config-vrf)# **show running-config**	Displays existing configuration information.

Caution

When you change label distribution mode, MP-BGP cancels all routes advertised in the VPN and advertises the routes again.

VRF-based label distribution mode is adopted by default. In this case, a PE first pops out the received packets with labels and then chooses routes to forward the packets based on the IP routing table.

Configuring Import and Export Policies for VPN Routes (Optional)

In most situations, you can define the route-target import attribute in VRF configuration mode to determine the routes to be imported into the VRF and define the route-target export attribute to determine the RTs to be carried in the routes. These configurations are effective for all routes. In certain application scenarios that require accurate policy-based control on the import and export of VPN routes, however, you need to adopt policies.

Enter privileged EXEC mode and use the following commands to configure import and export policies.

Command	Function
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **ip vrf** *vrf-name*	Creates a VRF and enters VRF configuration mode.

Ruijie(config-vrf)# **import map** *routemap-name*	Sets the policy to import VPNv4 routes from the remote end to the local VPN routes based on the rules defined in the route map.
Ruijie(config-vrf)# **export map** *routemap-name*	Sets the extended community attribute to export VPNv4 routes from the local end to the remote end based on the rules defined in the route map.
Ruijie(config-vrf)# **show running-config**	Displays existing configuration information.

Caution

The rules defined by using the **import map** command take effect after the import extended community attribute defined in the VRF. That is, VPN routes received from the remote end can be further filtered by using the rules defined with the **import map** command only after the routes match the extended community attribute defined by the **route-target import** command in the VRF.

# Configure a route map for importing VPN routes with the RT of 100:1 to vrf1.

```
Ruijie# configure terminal
Ruijie(config)# ip extcommunity-list 1 permit rt 100: 1
Ruijie(config)# route-map IN-RT-FILTER
Ruijie(config-route-map)# match extcommunity 1
Ruijie(config-route-map)# exit
Ruijie(config)# ip vrf vrf1
Ruijie(config-vrf)# rd 100: 2
Ruijie(config-vrf)# route-target export 100: 30
Ruijie(config-vrf)# import map IN-RT-FILTER
Ruijie(config-route-map)# end
Ruijie# show ip vrf detail vrf1
VRF vrf1: default RD : 100: 2
Interfaces:
Vlan 1 //Interface bound to the VRF
Export VPN route-target communities: //Export extended community attribute list
RT : 100: 30
No import VPN route-target community //Import extended community attribute list (not
configured)
import-map: IN-RT-FILTER//Import policy
```

Configuring a Static L3VPN FTN and ILM (Optional)

In most situations, MP-BGP assigns labels to private routes and a public LSP is generated by running LDP on a public network. You can also configure a static LSP to assign labels to private routes and set up private LSPs.

To configure an FTN for the L3VPN on the PE, enter privileged EXEC mode and use the following commands.

Command	Function
Ruijie# **config terminal**	Enters global configuration mode.

Ruijie(config)# **mpls static l3vpn-ftn** *vrf-name ip-address/mask* **out-label** *out-label* **remote-pe** *ip-address*	Configures a static private FTN that specifies the egress of the FEC as another PE. In this case, you must specify the private label and the egress PE. The address of the egress PE is then used to configure the public LSP.
Ruijie(config)# **mpls static l3vpn-ftn** *vrf-name fec-prefix/fec-mask* **local-forward nexthop** *interface-name nexthop-ip*	Configures a static private FTN that specifies the egress of the FEC as the local PE. In this case, you must specify the outgoing interface on the local PE and the next-hop address (the outgoing interface and the next hop are generally in another VRF). You can use this command when the local PE has several VRFs that belong to the same VPN.
Ruijie# **show running-config**	Displays existing configuration information.

To configure an ILM for the L3VPN on the PE, enter privileged EXEC mode and use the following commands.

Command	**Function**
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **mpls static ilm in-label** *in-label* **forward-action pop-l3vpn-nexthop** *vrf-name* **nexthop** *interface-name nexthop-ip-address* **fec** *ip-address/mask*	Configures an ILM entry for the L3VPN on the PE. You need to specify the incoming label, the outgoing interface, and the next-hop address.
Ruijie# **show running-config**	Displays existing configuration information.

Caution

The configured static private FTN and ILM take effect only after the corresponding public LSP is set up. To set up the public LSP, see the "Configuring an MPLS Network" section. You can set up a public LSP through LDP or static configurations.

Configuring an Inter-AS VPN

On an actual network, different sites of VPN users may be located on different ASs and mutual communication is required between these sites. In this case, the VPN routes should be exchanged between different ASs. This technology is called the inter-AS VPN.

RFC 4364 introduces three inter-AS VPN schemes:

- OptionA: VRF-to-VRF mode
- OptionB: single-hop MP-EBGP mode
- OptionC: multi-hop MP-EBGP mode

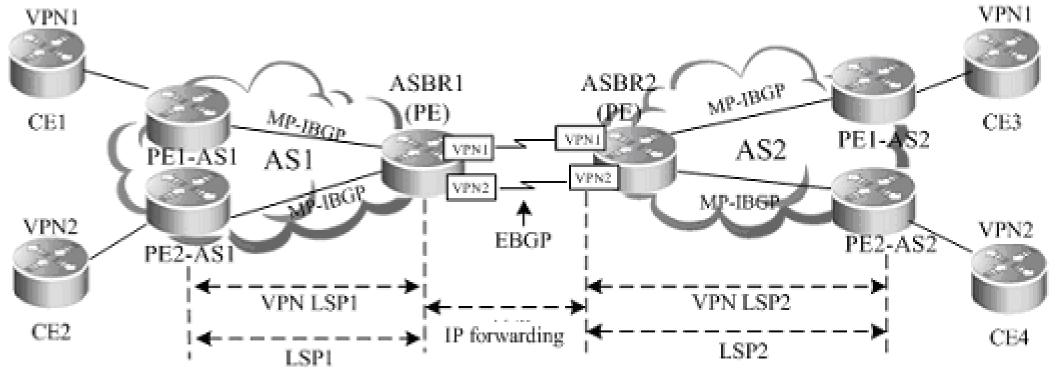
OptionA: VRF-to-VRF Mode

Also referred to as VRF back-to-back, VRF-to-VRF mode is easy to implement. The autonomous system boundary router (ASBR) of an AS sets up a VRF for each inter-AS VPN to bind an interface to the VRF. The VRFs on ASBRs then exchange VPN routes through the interface.

The purpose of creating a VRF and binding an interface to it is as follows:

- To receive VPN routes from the local AS
- To set up an EBGP connection between the VRF and the VRF of another AS to exchange IPv4 routes

Figure 8 VRF-to-VRF inter-AS VPN



As shown in the preceding figure, the VRFs between the ASBRs set up common EBGP sessions to exchange IPv4 routes and the ASBRs and PEs set up MP-IBGP sessions to exchange VPN routes. For the VRF on an ASBR, the other VRF, with which the EBGP session is set up, is equivalent to a CE. This configuration scheme is similar to the common intra-domain configuration scheme. The ASBRs and PEs set up MP-IBGP sessions to exchange VPN routes. The VRFs of ASBRs set up EBGP sessions in BGP VRF address family configuration mode to exchange IPv4 routes.

- Characteristics and limitations

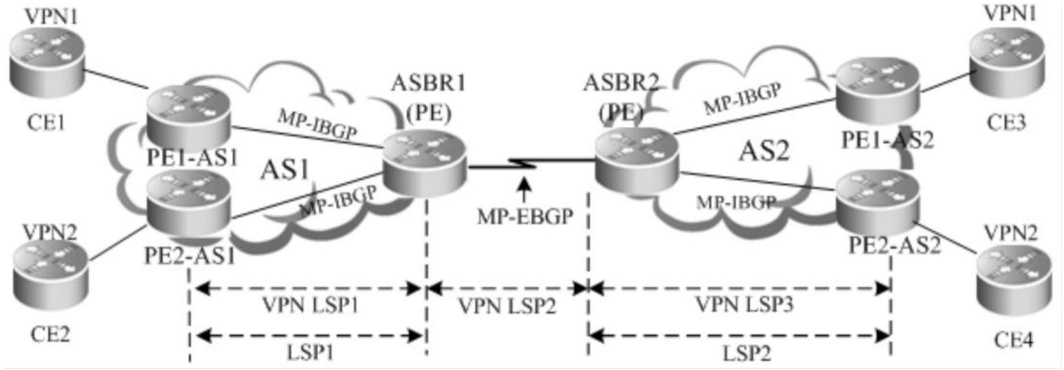
VRF-to-VRF mode is easy to implement by directly using MP-IBGP. The service deployment is also simple. This scheme, however, requires an interface (generally a logical sub-interface) for each inter-AS VPN on an ASBR. The number of bound interfaces at least should be equal to the number of inter-AS VPNs. You should configure an interface for each VPN on the ASBR, complicating network expansion. In addition, the separate creation of sub-interfaces for each VPN imposes high requirements on ASBRs. As a result, this scheme is generally applicable to networks with a small number of inter-AS VPNs.

The configuration of OptionA is similar to that of a BGP/MPLS IP VPN and is not described here.

OptionB: Single-Hop MP-EBGP Mode

In the OptionA scheme, you need to configure a VRF for each VPN on an ASBR and bind an interface to the VRF. This is because VPN routes cannot be directly transmitted between EBGP peers and can only be carried through MP-IBGP. If the VPN routes can be directly transmitted between EBGP peers, you are not required to configure VRFs on the ASBR. This is clearly a better implementation mode. In this case, the OptionB scheme extends MP-IBGP and allows the direct distribution of VPN routes between ASBRs. This is called single-hop EBGP, as shown in the following topology.

Figure 9 OptionB inter-AS VPN



- Characteristics and limitations

The advantage of this MP-EBGP scheme is that you are not required to configure a sub-interface for each site of VPN users on an ASBR. You are also not required to set up an inter-AS LSP. The VPN routes are directly transmitted between single-hop MP-EBGP neighbors. The VPN route information, however, is maintained and spread by the ASBRs between ASs. If a large number of VPN routes exist, the ASBRs are faced with heavy pressure. Because the ASBRs also generally assume tasks of forwarding IP packets on the public network, high requirements are imposed on these devices. In addition, the ASBRs cancel the RT filtering function for received VPN routes. The VPN routes on PEs may be spread to the ASBRs in another AS. This may lead to the leakage of VPN routes. As a result, the SPs, who exchange VPN routes, must reach trust agreements on route exchange. The ASBRs should trust each other and implement corresponding route filtering policies. The OptionB scheme is applicable to networks with lots of inter-AS VPN services.

OptionB has two schemes:

- The ASBR does not change the next hop of a VPN route.
- The ASBR changes the next hop of a VPN route.

The following describes the configuration procedures of the two schemes.

Scheme 1: Next Hop Unchanged

When an ASBR receives VPN routes sent from the ASBR in another AS and sends the routes to the MP-IBGP neighbors in the local AS, the next hop of the routes is not changed. This mode is called the "OptionB Next Hop Unchanged Scheme". In this mode, the PEs and ASBRs in an AS still set up MP-IBGP sessions to exchange VPN routes and the two ASBRs set up MP-EBGP sessions to directly exchange VPN routes. When sending routes to an MP-IBGP neighbor, the ASBR does not change the next hop of the VPN routes received from the MP-EBGP neighbor. This requires that the PE in the AS should have a route to the next hop address (that is, the ASBR in another AS). For this purpose, you can configure the local ASBR to redistribute routes destined for the other ASBR to the IGP protocol in the local AS. In this manner, the address of the ASBR in another AS becomes reachable and you can set up an LSP through LDP.

The configuration procedure is as follows:

11) Configuring route exchange between PEs and CEs

12) Configuring an IGP and MPLS signaling protocol in an AS

13) Configuring an ASBR to cancel the default RT filtering function

14) Configuring PEs and ASBRs in the same AS to exchange VPN route information

15) Setting up an MP-EBGP session between ASBRs

16) Configuring route map rules to filter VPN routers (optional)

17) Configuring an IGP to redistribute ASBR routes of another AS

- Configuring route exchange between PEs and CEs

This procedure is similar to configuring route exchange between PEs and CEs and is not described here.

- Configuring an IGP and MPLS signaling protocol in an AS

This procedure is similar to configuring an MPLS network and is not described here.

- Configuring an ASBR to cancel the default RT filtering function

By default, a PE rejects a VPN route sent by another PE (or ASBR), if the route is not imported by any VRF on the PE. Therefore, you should disable the default filtering on an ASBR so that the ASBR can receive all VPN routes from others PEs (or ASBRs), no matter whether these routes are imported into the local VRF or not.

Enter privileged EXEC mode and use the following commands.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **router bgp** *asn-number*	Enables BGP and enters BGP configuration mode.
Ruijie(config-router)# **no bgp default route-target filter**	Disables RT filtering.
Ruijie(config-router)# **show running-config**	Displays existing configuration information.

# Disable RT filtering.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 2
Ruijie(config-router)# no bgp default route-target filter
```

- Configuring PEs and ASBRs in the same AS to exchange VPN route information

This procedure is similar to configuring PEs to transmit VPN routes and is not described here.

- Setting up an MP-EBGP session between ASBRs

Set up directly-connected single-hop MP-EBGP sessions between inter-AS ASBRs to distribute VPN routes.

Enter privileged EXEC mode and use the following commands.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **router bgp** *asn-number*	Enables BGP and enters BGP configuration mode.
Ruijie(config-router)# **neighbor** *asbr-address* **remote-as** *asbr-asn-number*	Configures an ASBR EBGP session.
Ruijie(config-router)# **address-family vpnv4**	Enters the BGP VPN address family.
Ruijie(config-router-af)# **neighbor** *asbr-address* **activate**	Enables the VPN route exchange with the peer.
Ruijie(config-router-af)# **show running-config**	Displays existing configuration information.

# Configure an EBGP neighbor at 20.20.20.2 and activate the VPN address family.

```
Ruijie# configure terminal
```

```
Ruijie(config)# router bgp 2
Ruijie(config-router)# neighbor 20.20.20.2 remote-as 1
Ruijie(config-router)# address-family vpnv4
Ruijie(config-router-af)# neighbor 20.20.20.2 activate
```

⚠ Caution

You must run the **label-switching** command on the interface that connects two ASBRs to enable MPLS on the interface so that the links between the ASBRs can forward MPLS packets.

☑ On a router, use the **ip ref** command on the interface to enable fast forwarding to improve forwarding performance.

⚠ Caution

If the ASBRs do not use directly connected addresses to set up an MP-EBGP session but use the loopback address with a 32-bit mask as the source address to set up an MP-EBGP session, you must use the **neighbor ebgp-multihop** command to enable multi-hop EBGP. At the same time, you must configure static routes on the ASBR to the loopback address on the peer, enable LDP or configure a static FTN (with an outgoing label as 3, indicating that the ASBR is the second to last hop).

- Configuring route map rules to filter VPN routes (optional)

In view of the AS security in actual applications, you can generally configure policies on ASBRs to send or receive only certain VPN routers. You can realize this purpose by filtering the RT extended community attributes of VPN routes. In addition, all VPN routes are saved because the default RT filtering function is disabled on the ASBR. In this case, you can configure VPN routing policies to receive only inter-AS VPN routes sent from the local AS, lessening the capacity pressure of the ASBR.

To configure a filtering policy, enter privileged EXEC mode and use the following commands.

Command	**Function**
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **ip extcommunity-list standard** *extcommunity-name* | *extcommunity-number* {**permit|deny**} **rt** *rt-value*	Creates a rule for the extended community attribute list.
Ruijie(config)# **show ip extcommunity-list** [*list-number*|*list-name*]	Verifies the configured rule for the extended community attribute list.
Ruijie(config)# **route-map** *route-map-name* **permit** [*number*]	Creates a route map rule and enters route map configuration mode.
Ruijie(config-route-map)# **match extcommunity** *extcommunity-name*|*extcommunity-number*	Sets the RT matching rule for a route map.
Ruijie(config-route-map)# **show route-map** route-map-name	Displays the route map rule.
Ruijie(config-route-map)# **exit**	Exits route map configuration mode.
Ruijie(config)# **router bgp** *as-num*	Enables BGP and enters BGP configuration mode.
Ruijie(config-router)# **address-family vpnv4**	Enters the VPN address family.

Command	Function
Ruijie(config-router-af)# **neighbor** *peer-address* **route-map** *route-map-name* **in**	Filters the VPN routes received from the ASBR in another AS.
Ruijie(config-router-af)# **neighbor** *peer-address* **route-map** *route-map-name* **out**	Filters the VPN routes sent to the ASBR in another AS.
Ruijie(config-router-af)# **show running-config**	Displays existing configuration information.

# Configure an ASBR to receive VPN routes with an RT value of 100:1 from the MP-IBGP peer at 1.1.1.1.

```
Ruijie# configure terminal
Ruijie(config)# ip extcommunity-list standard RT permit rt 100:1
Ruijie(config)# show ip extcommuniy-list RT
Named extended community standard list RT
permit rt 100:1
Ruijie(config)# route-map RT-IN permit
Ruijie(config-route-map)# match extcommunity RT
Ruijie(config-route-map)# show route-map RT-IN
route-map map, permit, sequence 10
  Match clauses:
  extcommunity (extcommunity-list filter):RT
  Set clauses:
  Policy routing matches: 0 packets, 0 bytes
Ruijie(config-route-map)# exit
Ruijie(config)# router bgp 100
Ruijie(config-router)# neighbor 1.1.1.1 remote-as 100
Ruijie(config-router)# neighbor 1.1.1.1 update-source loopback 0
Ruijie(config-router)# address-family vpnv4
Ruijie(config-router-af)# neighbor 1.1.1.1 activate
Ruijie(config-router-af)# neighbor 1.1.1.1 route-map RT-IN in
Ruijie(config-router-af)# end
```

- Configuring an IGP to redistribute ASBR routes of another AS

Because the ASBR does not change the next hop of VPN routes sent to the IBGP peer, the next hop address of VPN routes learned by the PEs in the local AS is the ASBR address in another AS. Therefore, you must configure the PEs to learn the route to the next hop address. For the single-hop directly-connected MP-EBGP session where BGP is enabled to carry labels (through IPv4 routes or VPN routes), MP-BGP supports the automatic generation of a host route with a 32-bit mask and FTN entry (with the outgoing label 3) on the ASBR. In this manner, the tunnel egress is not terminated on the local ASBR. Therefore, as long as the ASBR redistributes the host route to the IGP in the local AS, the PEs can learn routes to the ASBR in the other AS.

Enter privileged EXEC mode and use the following commands.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **router** *igp*	Enables an IGP that can be OSPF, RIP, or IS-IS.
Ruijie(config-router)# **redistribute connected subnets**	Redistributes directly connected subnet routes.

Ruijie(config-router)# **show running-config**	Displays existing configuration information.

# Configure OSPF on an ASBR to redistribute the directly connected subnet routes.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 1
Ruijie(config-router)# redistribute connected subnets
```

Scheme 2: Next Hop Changed

When an ASBR receives VPN routes sent from the ASBR in another AS and sends the routes to the PEs in the local AS, the next hop of the routes is changed. This mode is called the "OptionB Next Hop Changed Scheme". In this mode, the PEs and ASBRs in the same AS can set up MP-IBGP sessions to exchange VPN routes. Two ASBRs can set up MP-EBGP sessions to exchange VPN routes. Upon receipt of a VPN route from another ASBR neighbor, an ASBR changes the next hop to its own address when advertising the route to the MP-IBGP peer in the AS.

The configuration procedure is as follows:

18) Configuring route exchange between PEs and CEs
19) Configuring an IGP and MPLS signaling protocol in an AS
20) Configuring an ASBR to cancel the default RT filtering function
21) Setting up an MP-IBGP session between an ASBR and PE and changing the next hop address to its own address
22) Setting up an MP-EBGP session between ASBRs
23) Configuring route map rules to filter VPN routes (optional)

- Configuring route exchange between PEs and CEs

This procedure is similar to configuring route exchange between PEs and CEs and is not described here.

- Configuring an IGP and MPLS signaling protocol in an AS

This procedure is similar to configuring an MPLS network and is not described here.

- Configuring an ASBR to cancel the default RT filtering function

This procedure is similar to configuring an ASBR to cancel the default RT filtering function in Scheme 1 and is not described here.

- Setting up an MP-IBGP session between an ASBR and PE and changing the next hop address to its own address

By default, an ASBR does not change the next hop of the VPN route received from an MP-EBGP peer when the ASBR sends the route to the MP-IBGP peer. You can configure the ASBR to forcibly change the next hop of the VPN route to the ASBR address in the local AS. In this manner, the PEs in the local AS are not required to learn the address of the peer ASBR. This is the major difference with Scheme 1 (Next Hop Unchanged Scheme).

Enter privileged EXEC mode and use the following commands.

Command	**Function**
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **router bgp** *asn-num*	Enables BGP and enters BGP configuration mode.

Ruijie(config-router)# **neighbor** *pe-address* **remote-as** *asn-num*	Sets up an IBGP session with a PE.
Ruijie(config-router)# **neighbor** *pe-address* **update-source** *interface-name*	Specifies the local loopback interface as the source address to set up an IBGP session.
Ruijie(config-router)#**address-family vpnv4**	Enters BGP VPN address family configuration mode.
Ruijie(config-router-af)# **neighbor** *pe-address* **activate**	Enables the VPN route exchange with the peer.
Ruijie(config-router-af)# **neighbor** *pe-address* **next-hop-self**	Sets the ASBR to change the next hop to its own address when sending VPN routes to the IBGP neighbor.
Ruijie(config-router-af)# **show running-config**	Displays existing configuration information.

# Set up an MP-IBGP session, activate the VPN address family, and change the next hop address to the ASBR address.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 1.1.1.1 remote-as 1
Ruijie(config-router)# neighbor 1.1.1.1 update-source loopback 0
Ruijie(config-router)# address-family vpnv4
Ruijie(config-router-af)# neighbor 1.1.1.1 activate
Ruijie(config-router-af)# neighbor 1.1.1.1 next-hop-self
Ruijie(config-router-af)# end
```

- Setting up an MP-EBGP session between ASBRs

This procedure is similar to setting up an MP-EBGP session between ASBRs in Scheme 1 and is not described here.

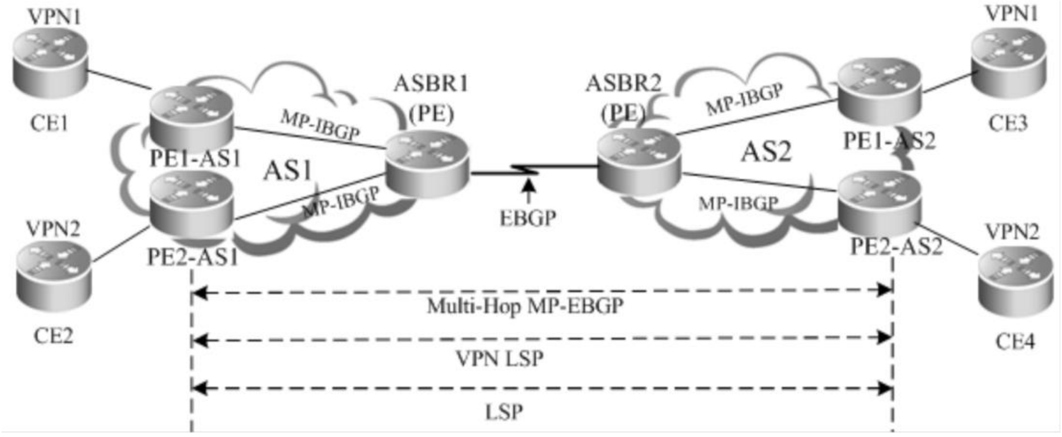
- Configuring route map rules to filter VPN routes (optional)

This procedure is similar to configuring route map rules to filter VPN routers (optional) in Scheme 1 and is not described here.

OptionC: Multi-Hop MP-EBGP Mode

Both OptionA and OptionB can meet the networking requirements of inter-AS VPNs. In these two schemes, ASBRs are required to maintain and advertise VPN routes. If a large number of inter-AS VPN routes should be advertised in each AS, the ASBRs may become the bottleneck of further network expansion. To solve this problem, a third scheme is developed, that is, the multi-hop MP-EBGP. In multi-hop MP-EBGP mode, the PEs in different ASs set up multi-hop MP-EBGP sessions to directly exchange VPN routes. As a result, the ASBRs are not required to maintain or advertise VPN routes.

Figure 10 Multi-hop MP-EBGP



- Characteristics and limitations

In multi-hop MP-EBGP mode, only PEs rather than ASBRs are required to store VPN information. This incurs complex configurations. This scheme is applicable to networks to be deployed with inter-AS VPN services on a large scale.

In terms of implementation principle, OptionC is further classified into two modes:

24) Enabling label distribution for IPv4 routes only between EBGP neighbors.

25) Enabling label distribution for IPv4 routes between EBGP and IBGP neighbors.

To facilitate expansion in OptionC, each AS is generally deployed with a route reflector (RR). The RRs of two ASs set up multi-hop MP-EBGP sessions to exchange VPN routes. Judged from deployment, OptionC can be referred to as the scheme of "Setting Up a Multi-Hop MP-EBGP Session Between RRs".

The following describes the configuration procedures of these schemes.

Scheme 1: Enabling Label Distribution for IPv4 Routes Only Between EBGP Neighbors

In this scheme, the IGP (such as OSPF or RIP) that runs on an ASBR is required to redistribute BGP routes so that each device in the AS can have routes to the PE in another AS. In the AS, you can use LDP to assign labels to the routes to the PE in another AS and set up an LSP. On the directly connected ASBRs of the two ASs, enable label distribution for IPv4 routes. In this manner, BGP serves as the MPLS signaling protocol to assign labels to the routes to the PE in another AS and set up an inter-AS LSP.

The configuration procedure is as follows:

26) Configuring route exchange between PEs and CEs in each AS

27) Configuring an IGP and MPLS signaling protocol in an AS

28) Setting up an EBGP session between ASBRs to distribute labels for IPv4 routes

29) Configuring an ASBR to redistribute inter-AS PE routes learned from EBGP to the IGP

30) Configuring a multi-hop MP-EBGP session

- Configuring route exchange between PEs and CEs

This procedure is similar to configuring route exchange between PEs and CEs and is not described here.

- Configuring an IGP and MPLS signaling protocol in an AS

This procedure is similar to configuring an MPLS network and is not described here.

- Setting up an EBGP session between ASBRs to enable label distribution for IPv4 routes

Set up an EBGP session between inter-AS ASBRs and enable label distribution of IPv4 routes. To import PE routes to BGP, you can use the **network** command in BGP IPv4 address family configuration mode or run commands to redistribute IGP routes. In view of the AS security in actual applications, you are generally required to configure IPv4 route distribution policies on ASBRs. By configuring route map rules, you can control the routes sent to neighbors and specify whether the routes carry labels. Similar control is available for receiving routes.

Enter privileged EXEC mode and use the following commands.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **router bgp** *asn-num*	Enables BGP and enters BGP configuration mode.
Ruijie(config-router)# **neighbor** *asbr-address* **remote-as** *asbr-asn-num*	Sets up an EBGP session with an ASBR.
Ruijie(config-router)#**address-family ipv4**	Enters BGP IPv4 address family configuration mode.
Ruijie(config-router-af)# **neighbor** asbr-address **send-label**	Configures the device to exchange labeled IPv4 routes with the ASBR peer in another AS.
Ruijie(config-router-af)# **network** *pe-address* **mask** *mask*	(Optional) Configures PE addresses to be imported into the BGP routing table in the local AS, that is, host routes of each PE in the AS.
Ruijie(config-router-af)# **neighbor** *asbr-address* **route-map** *routemap-name* **out**	(Optional) Configures a route distribution policy to control the routes sent to neighbors and specify whether the routes can carry labels, by defining a route map rule.
Ruijie(config-router-af)# **neighbor** *asbr-address* **route-map** *routemap-name* **in**	(Optional) Configures a route distribution policy to receive only labeled routes by defining a route map rule.
Ruijie(config-router-af)# **show running-config**	Displays existing configuration information.

⚠ Caution

You must run the **label-switching** command on the interface that connects two ASBRs to enable MPLS on the interface so that the links between the ASBRs can forward MPLS packets.

☑ On a router, use the **ip ref** command on the interface to enable fast forwarding to improve forwarding performance.

⚠ Caution

If the ASBRs do not use directly connected addresses to set up an MP-EBGP session but use the loopback address with a 32-bit mask as the source address to set up an MP-EBGP session, you must use the **neighbor ebgp-multihop** command to enable multi-hop EBGP. At the same time, you must configure static routes on the ASBR to the loopback address on the peer, enable LDP or configure a static FTN (with an outgoing label as 3, indicating that the ASBR is the second to last hop).

# Set up an EBGP session between ASBRs, enable label distribution for IPv4 routes, and run the **network** command to import PE routes to BGP.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 20.20.20.2 remote-as 2
Ruijie(config-router)# address-family ipv4
Ruijie(config-router-af)# neighbor 20.20.20.2 send-label
Ruijie(config-router-af)# network 10.10.10.10 mask 255.255.255.255
Ruijie(config-router-af)# end
```

In actual applications, an ASBR is generally required to distribute labels for PE routes for only inter-AS VPN services. For this purpose, you can use the route map rules defined by the **set mpls-label** command.

The **set mpls-label** command sets labels for routes. You can create a route map rule to distribute only inter-AS PE routes to the peer ASBR and set labels for the routes. Set route map rules and then run the **neighbor** *peer-address* **route-map** *rmap_name* **out** command in BGP IPv4 address family configuration mode to associate the rules with the route map.

The following example creates a route map to assign an MPLS label to the route with a prefix as 1.1.1.1/32, to assign a common IPv4 route without a label to the route with a prefix as 1.1.1.2/32, and not to send neighbors routes that fail to match acl1 and acl2.

```
Router(config)# ip access-list standard acl1
Router(config-std-nacl)# permit host 1.1.1.1
Router(config-std-nacl)# exit
Router(config)# ip access-list standard acl2
Router(config-std-nacl)# permit host 1.1.1.2
Router(config-std-nacl)# exit
Router(config)# route-map out-as permit 10
Router(config-route-map)# match ip address acl1
Router(config-route-map)# set mpls-label
Router(config-std-nacl)# exit
Router(config)# route-map out-as permit 20
Router(config-route-map)# match ip address acl2
Router(config)# router bgp 100
Router(config-router)# neighbor 30.30.30.2 remote-as 100
Router(config-router)# neighbor 30.30.30.2 route-map out-as out
```

Similarly, to receive only labeled IPv4 routes, you can run the **match mpls-label** command in route map mode. Set route map rules and then run the **neighbor** *peer-address* **route-map** *rmap_name* **in** command to associate the rules with the route map.

The following example creates a route map to receive labeled IPv4 routes from only the BGP peer at 30.30.30.2 and reject other routes.

```
Router(config)# route-map match-mpls
Router(config-route-map)# match mpls-label
Router(config)# router bgp 100
Router(config-router)# neighbor 30.30.30.2 remote-as 100
Router(config-router)# neighbor 30.30.30.2 route-map match-mpls in
```

- Configuring an ASBR to redistribute inter-AS PE routes learned from EBGP to the IGP

When an ASBR learns a route to the PE in another AS from the peer ASBR, the ASBR should advertise the route to other PEs in the local AS. The ASBR should also set up an LSP to the PE in another AS. In this manner, the ASBR can redistribute routes learned from EBGP to the IGP and at the same time, enable LDP to distribute labels for BGP routes and then set up an LSP to the PE in another AS.

Enter privileged EXEC mode and use the following commands.

Command	**Function**
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **router** *igp*	Enters IGP configuration mode.
Ruijie(config-router)# **redistribute bgp subnets [route-map** *routemap-name*]	Redistributes BGP routes. Route filtering by using route map rules is optional.
Ruijie(config-router)# **exit**	Exits IGP configuration mode.
Ruijie(config)# **mpls router ldp**	Enters LDP configuration mode.
Ruijie(config-mpls-router)# **ldp router-id interface loopback** *0* force	Configures the LDP router ID. The loopback address is generally used as the router ID.
Ruijie(config-mpls-router)# **advertise-labels for bgp-routes [acl** *acl-name*]	Distributes labels for BGP routes. Filtering based on ACL rules is optional.
Ruijie(config-mpls-router)# **show running-config**	Displays existing configuration information.

Caution

By default, LDP distributes labels for only IGP routes and does not distribute labels for BGP routes. To distribute labels for BGP routes, you can run the **advertise-labels for bgp-routes** command.

# Configure an IGP and MPLS signaling protocol in an AS.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 1
Ruijie(config-router)# redistribute bgp subnets
Ruijie(config-router)# exit
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# advertise-labels for bgp-routes
Ruijie(config-mpls-router)# end
```

When an IGP redistributes the learned BGP routes in the OptionC scheme, you can run the **redistribute bgp subnets route-map** *routemap-name* command in IGP configuration mode to control the BGP routes to be redistributed to the IGP. In LDP configuration mode, you can run the **advertise-labels for bgp-routes acl** *acl-name* command to control the labels distributed for BGP routes.

# Configure ACL rules and route map routes so that:

- The IGP redistributes only routes 1.1.1.1 and 2.2.2.2.
- The LDP assigns labels to only routes 1.1.1.1 and 2.2.2.2.

The configuration procedure is as follows:

```
Router(config)# ip access-list extended 101
Router(config-ext-nacl)# permit ip host 1.1.1.1 any
Router(config-ext-nacl)# permit ip host 2.2.2.2 any
Router(config-ext-nacl)# exit
Router(config)# route-map pe-routes
Router(config-route-map)# match ip address 101
Router(config-route-map)# exit
Router(config)# router ospf 1
Router(config-router)# redistribute bgp subnets route-map pe-routes
Router(config-route-map)# exit
Router(config)# mpls router ldp
Router(config-mpls-router)# advertise-labels for bgp-routes acl 101
```

■ Configuring a multi-hop MP-EBGP session

In the earlier steps, the inter-AS LSP is already set up. At this time, you can directly set up a multi-hop MP-EBGP session on the PE to be deployed with inter-AS VPN services with the PE in another AS. The session can then exchange VPN routes.

To configure a multi-hop MP-EBGP session, enter privileged EXEC mode and use the following commands.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **router bgp** *asn-num*	Enables BGP and enters BGP configuration mode.
Ruijie(config-router)# **neighbor** *ebgp-peer-address* **remote-as** *ebgp-asn-num*	Sets up a multi-hop EBGP session with the PE in another AS.
Ruijie(config-router)# **neighbor** *ebgp-peer-address* **update-source** *interface-name*	Configures the device to use the loopback address to set up a neighbor relation with the EBGP peer.
Ruijie(config-router)# **neighbor** *ebgp-peer-address* **ebgp-multihop**	Configures multi-hop attributes.
Ruijie(config-router)#**address-family vpnv4**	Enters BGP VPN address family configuration mode.
Ruijie(config-router-af)# **neighbor** *ebgp-peer-address* **activate**	Enables the VPN route exchange with the peer.
Ruijie(config-router-af)# **exit**	Exits the BGP VPN address family.
Ruijie(config-router)# **address-family ipv4**	Enters BGP IPv4 address family configuration mode.
Ruijie(config-router-af)# **no neighbor** *ebgp-peer-address* **activate**	Disables the IPv4 route exchange.
Ruijie(config-router)# **show running-config**	Displays existing configuration information.

Caution

The exchange of IPv4 routes is not required in a multi-hop MP-EBGP session. At least the routes of the two addresses used to set up the BGP session should be avoided. Otherwise, a PE has two routes to the PE in another AS. One route is advertised by the ASBR in the local AS and the other is advertised by the multi-hop EBGP session. According to the BGP specification, the EBGP route has a higher priority over the IBGP route by default. As a result, BGP chooses the route advertised by the multi-hop BGP, resulting in continuous flapping of routes on the PE to the PE in another AS. The VPN routes are thus not reachable.

# Set up a multi-hop EBGP session.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 1.1.1.1 remote-as 2
Ruijie(config-router)# neighbor 1.1.1.1 update-source loopback 0
Ruijie(config-router)# neighbor 1.1.1.1 ebgp-multihop
Ruijie(config-router)# address-family vpnv4
Ruijie(config-router-af)# neighbor 1.1.1.1 activate
Ruijie(config-router-af)# exit
Ruijie(config-router)# address-family ipv4
Ruijie(config-router-af)# no neighbor 1.1.1.1 activate
Ruijie(config-router-af)# end
```

Scheme 2: Enabling Label Distribution for IPv4 Routes Between EBGP and IBGP Neighbors

In Scheme 1 (Enabling Label Distribution for IPv4 Routes Only Between EBGP Neighbors), the IGP and LDP in one AS are required to maintain the PE routes from another AS. That is, inter-AS PE routes should be advertised to each device in the AS. In view of the AS security in actual applications, the PE routes of another AS are generally not advertised to each device in the local AS. Instead, these routes should be owned by the BGP protocol so that the routes can be transparent to the IGP and LDP in the local AS. You can enable label distribution for IPv4 routes between EBGP and IBGP neighbors.

This scheme differs from Scheme 1 in that the IGP on an ASBR is not required to redistribute BGP routes and that the LDP is not required to assign labels to BGP routes, though the LDP is still responsible for the setup of an LSP in the local AS. The setup of an inter-AS LSP, however, requires label distribution for IPv4 routes between both IBGP and EBGP neighbors. The PEs are also required to push three consecutive layers of labels.

The configuration procedure is as follows:

31) Configuring route exchange between PEs and CEs in each AS

32) Configuring an IGP and MPLS signaling protocol in an AS

33) Setting up an IBGP session between a PE and an ASBR to distribute labels for IPv4 routes

34) Setting up an EBGP session between ASBRs to distribute labels for IPv4 routes

35) Configuring a multi-hop MP-EBGP session

■ Configuring route exchange between PEs and CEs

This procedure is similar to configuring route exchange between PEs and CEs and is not described here.

■ Configuring an IGP and MPLS signaling protocol in an AS

This procedure is similar to configuring an MPLS network and is not described here.

■ Setting up an IBGP session between a PE and an ASBR to distribute labels for IPv4 routes

This scheme differs from Scheme 1 mainly in this configuration procedure. In this scheme, the PE routes that are learned by EBGP from another AS are not redistributed to the IGP in the local AS. Instead, the IBGP session between an ASBR and a PE is used to transmit the PE routes of another AS and BGP is used to assign labels to the PE routes.

Enter privileged EXEC mode and use the following commands.

Command	**Function**
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **router bgp** *asn-number*	Enables BGP and enters BGP configuration mode.
Ruijie(config-router)# **neighbor** *peer-address* **remote-as** *asn-number*	Sets up an IBGP session with an ASBR (PE).
Ruijie(config-router)# **neighbor** *peer-address* **update-source** *interface-name*	Configures the device to use the loopback address as the source address to set up the BGP session with an ASBR (PE) peer.
Ruijie(config-router)# **address-family ipv4**	Enters the IPv4 address family.
Ruijie(config-router-af)# **neighbor** *peer-address* **send-label**	Configures the device to exchange labeled IPv4 routes with an ASBR (PE) peer.
Ruijie(config-router-af)# **show running-config**	Displays existing configuration information.

Caution

Before you enable label distribution for IPv4 routes for an IBGP session with an IBGP peer, run the **neighbor update-source** command to specify the source address of the IBGP session. This source address must be the address of the loopback interface; otherwise, the inter-AS LSP cannot be set up.

# Configure a PE to set up an MP-IBGP session with the ASBR at 10.10.10.2.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 10.10.10.2 remote-as 1
Ruijie(config-router)# neighbor 10.10.10.2  update-source loopback 0
Ruijie(config-router)# address-family ipv4
Ruijie(config-router-af)# neighbor 10.10.10.2 activate
Ruijie(config-router-af)# neighbor 10.10.10.2 send-label
Ruijie(config-router-af)# exit
```

# Configure an ASBR to set up an MP-IBGP session with the PE at 10.10.10.1 in the local AS.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 10.10.10.1 remote-as 1
Ruijie(config-router)# neighbor 10.10.10.1 update-source loopback 0
Ruijie(config-router)# address-family ipv4
Ruijie(config-router-af)# neighbor 10.10.10.1 send-label
Ruijie(config-router-af)# end
```

- Setting up an EBGP session between ASBRs to enable label distribution for IPv4 routes

This procedure is similar to the corresponding procedure in Scheme 1 and is not described here.

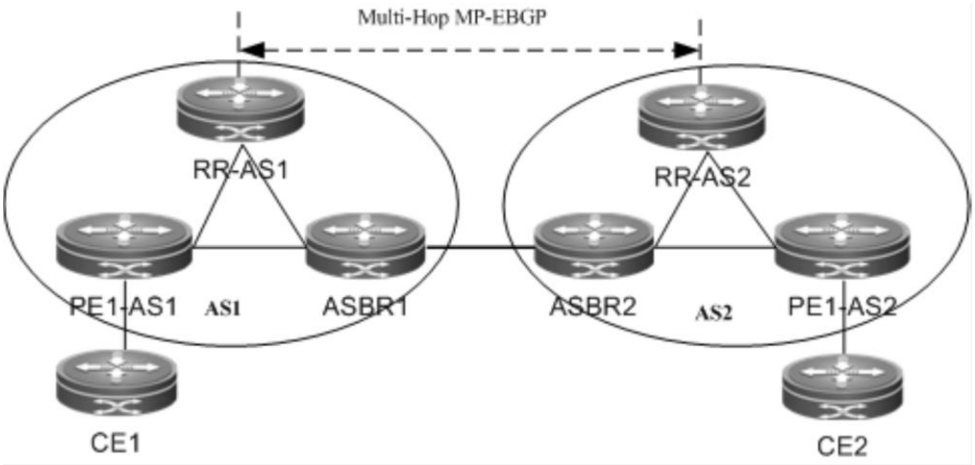
- Configuring a multi-hop MP-EBGP session

This procedure is similar to the corresponding procedure in Scheme 1 and is not described here.

Scheme 3: Setting Up a Multi-Hop MP-EBGP Session Between RRs

In the traditional OptionC scheme, the inter-AS VPN sites should be connected in full mesh mode. The addition of a single VPN site requires the setup of MP-MBGP connections with the PEs in other ASs, hindering the expansion of VPN sites. In this case, you can deploy an RR in each AS to solve this problem. Set up multi-hop MP-EBGP sessions between the RRs to exchange VPN routes.

Figure 11 Setting up a multi-hop MP-EBGP session between RRs in OptionC mode



As shown in the preceding figure, the RRs in the two ASs set up a multi-hop MP-EBGP session to exchange VPN routes. The configuration procedure is as follows:

36) Configuring route exchange between PEs and CEs in each AS

37) Configuring an IGP and MPLS signaling protocol in an AS

38) Setting up an MP-IBGP session between the RR and the PE and enabling label distribution for IPv4 routes

39) Setting up an IBGP session between the RR and the ASBR and enabling label distribution for IPv4 routes

40) Setting up an EBGP session between ASBRs to distribute labels for IPv4 routes

41) Configuring a multi-hop MP-EBGP session

- Configuring route exchange between PEs and CEs

This procedure is similar to configuring route exchange between PEs and CEs and is not described here.

- Configuring an IGP and MPLS signaling protocol in an AS

This procedure is similar to configuring an MPLS network and is not described here.

- Setting up an MP-IBGP session between the RR and the PE and enabling label distribution for IPv4 routes

Configure a PE to set up an MP-IBGP session with the RR to transmit VPN routes. At the same time, enable label distribution for IPv4 routes for the session.

Enter privileged EXEC mode and use the following commands.

Command	**Function**
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **router bgp** *asn-number*	Enables BGP and enters BGP configuration mode.
Ruijie(config-router)# **neighbor** *peer-address* **remote-as** *asn-number*	Sets up the IBGP session.
Ruijie(config-router)# **neighbor** *peer-address* **update-source** *interface-name*	Uses the address of the loopback address as the source address to set up an IBGP session.
Ruijie(config-router)# **address-family ipv4**	Enters the IPv4 address family.
Ruijie(config-router-af)# **neighbor** *peer-address* **Activate**	Enables IPv4 route exchange.
Ruijie(config-router-af)# **neighbor** *peer-address* **send-label**	Enables label distribution for IPv4 routes.
Ruijie(config-router-af)# n**eighbor** *peer-address* **route-reflector-client**	Configures all PE peers as the clients of the IPv4 RR.
Ruijie(config-router-af)# **exit**	Exits the IPv4 address family.
Ruijie(config-router)# **address-family vpnv4**	Enters the VPN address family.
Ruijie(config-router-af)# **neighbor** *peer-address* **Activate**	Enables the VPN route exchange with the peer.
Ruijie(config-router-af)# **neighbor** *peer-address* **route-reflector-client**	Configures all PE peers as the clients of the VPN RR.
Ruijie(config-router-af)# **show running-config**	Displays existing configuration information.

# Set up an MP-IBGP session between the RR and the PE. The configuration on the RR is as follows:

```
Ruijie# configure terminal
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 10.10.10.1 remote-as 1
Ruijie(config-router)# neighbor 10.10.10.1 update-source loopback 0
Ruijie(config-router)# address-family ipv4
Ruijie(config-router-af)# neighbor 10.10.10.1 activate
Ruijie(config-router-af)# neighbor 10.10.10.1 send-label
Ruijie(config-router-af)# neighbor 10.10.10.1 route-reflector-client
Ruijie(config-router-af)# exit
Ruijie(config-router)# address-family vpnv4
Ruijie(config-router-af)# neighbor 10.10.10.1 activate
Ruijie(config-router-af)# neighbor 10.10.10.1 route-reflector-client
Ruijie(config-router-af)# end
```

- Setting up an IBGP session between the RR and the ASBR and distributing labels for IPv4 routes

Set up an MP-IBGP session between the ASBR and the RR to receive routes from the RR to the PEs in the local AS and send routes from the RR to the PEs in another AS. At the same time, enable label distribution for IPv4 routes for the session.

Enter privileged EXEC mode and use the following commands.

Command	**Function**
Ruijie# config terminal	Enters global configuration mode.
Ruijie(config)# **router bgp** *asn-number*	Enables BGP and enters BGP configuration mode.
Ruijie(config-router)# **neighbor** *peer-address* **remote-as** *asn-number*	Sets up the IBGP session.
Ruijie(config-router)# **neighbor** *peer-address* **update-source** interface-name	Uses the address of the loopback address as the source address to set up an IBGP session.
Ruijie(config-router)# **address-family ipv4**	Enters the IPv4 address family.
Ruijie(config-router-af)# **neighbor** *peer-address* **activate**	Enables IPv4 route exchange.
Ruijie(config-router-af)# **neighbor** *peer-address* **send-label**	Enables label distribution for IPv4 routes.
Ruijie(config-router)# **show running-config**	Displays existing configuration information.

Note For the IBGP session between an RR and an ASBR, you are generally not required to set the ASBR as the client of the RR unless the ASBR also serves as a PE.

# Set up an IBGP session between the RR and the ASBR. The configuration on the RR (the configuration on the ASBR is similar) is as follows:

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 10.10.10.2 remote-as 1
Ruijie(config-router)# neighbor 10.10.10.2 update-source loopback 0
Ruijie(config-router)# address-family ipv4
Ruijie(config-router-af)# neighbor 10.10.10.2 activate
Ruijie(config-router-af)# neighbor 10.10.10.2 send-label
Ruijie(config-router-af)# end
```

- Setting up an EBGP session between ASBRs to enable label distribution for IPv4 routes

This procedure is similar to the corresponding procedure in Scheme 1 and is not described here.

- Configuring a multi-hop MP-EBGP session

Set up a multi-hop MP-EBGP session between the RRs of two ASs to exchange inter-AS VPN routes. At the same time, disable the transmission of IPv4 routes for the session. The PE routes are advertised to another AS through the ASBR.

Enter privileged EXEC mode and use the following commands.

Command	**Function**
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **router bgp** *asn-number*	Enables BGP and enters BGP configuration mode.
Ruijie(config-router)# **neighbor** *rr-address* **remote-as** ebgp-asn-numbe	Sets up the EBGP session.

Ruijie(config-router)# **neighbor** *rr-address* **update-source** *interface-name*	Uses the address of the loopback address as the source address to set up an EBGP session.
Ruijie(config-router)# **neighbor** *rr-address* **ebgp-multihop**	Configures multi-hop EBGP attributes.
Ruijie(config-router)# **address-family ipv4**	Enters the IPv4 address family.
Ruijie(config-router-af)# **no neighbor** *rr-address* **activate**	Disables IPv4 route exchange for the session.
Ruijie(config-router-af)# **exit**	Exits the IPv4 address family.
Ruijie(config-router)# **address-family vpnv4**	Enters the VPN address family.
Ruijie(config-router-af)# **neighbor** *rr-address* **Activate**	Enables the device to exchange VPN routes with the RR in another AS.
Ruijie(config-router-af)# **neighbor** *rr-address* **next-hop-unchanged**	(Optional) Configures the device not to change the next hop when advertising VPN routes to the peer.
Ruijie(config-router)# **show running-config**	Displays existing configuration information.

Caution

By default, the device changes the next hop of a route to its own address when advertising the route to an EBGP peer. Upon receipt of the VPN route, the PE site in another AS considers the next hop of the route as the RR. As a result, all inter-AS VPN traffic is transmitted through the RR. This is generally not the optimal forwarding path and has high requirements on the forwarding performance of the RR. To avoid the preceding situation, you can run the **neighbor next-hop-unchanged** command in VPNv4 address family configuration mode to configure the device not to change the next hop of a VPNv4 route sent to the BGP peer when you set up a multi-hop MP-EBGP session on the RR.

The exchange of IPv4 routes is not required in a multi-hop MP-EBGP session. At least the routes of the two addresses used to set up the BGP session should be avoided. Otherwise, a PE has two routes to the PE in another AS. One route is advertised by the ASBR in the local AS and the other is advertised by the multi-hop EBGP session. According to the BGP specification, the EBGP route has a higher priority over the IBGP route by default. As a result, BGP chooses the route advertised by the multi-hop BGP, resulting in continuous flapping of routes on the PE to the PE in another AS. The VPN routes are thus not reachable.

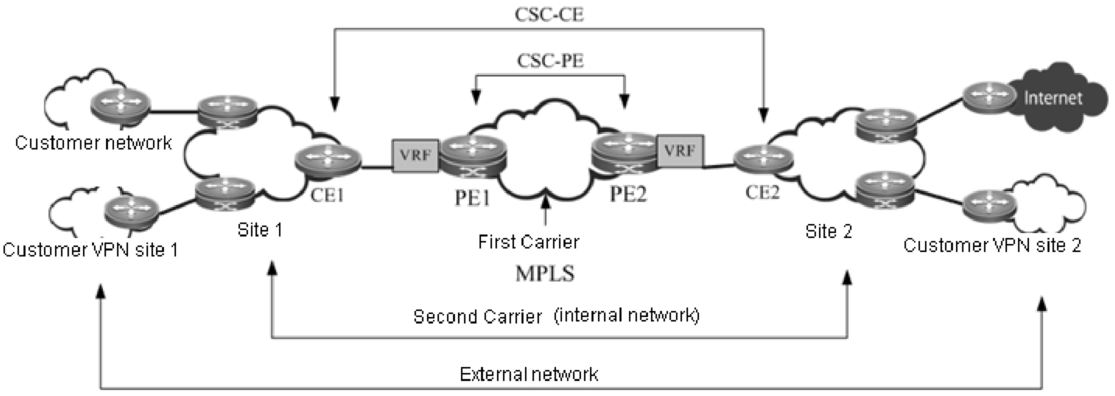
# Configure an RR to set up a multi-hop MP-EBGP session with the RR in another AS.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 30.30.30.2 remote-as 2
Ruijie(config-router)# neighbor 30.30.30.2 update-source loopback 0
Ruijie(config-router)# neighbor 30.30.30.2 ebgp-multihop
Ruijie(config-router)# address-family ipv4
Ruijie(config-router-af)# no neighbor 30.30.30.2 activate
Ruijie(config-router-af)# exit
Ruijie(config-router)# address-family vpnv4
Ruijie(config-router-af)# neighbor 30.30.30.2 activate
Ruijie(config-router-af)# neighbor 30.30.30.2 next-hop-unchanged
Ruijie(config-router-af)# end
```

Configuring Carrier's Carrier (CSC)

In a basic MPLS VPN, each site is a traditional IP network with a simple network structure. However, in there are some special VPN users. For example, a VPN user is also a service provider who leases the VPN service of an MPLS VPN service provider and then provides specific services for users. In this case, the MPLS VPN service provider is called a provider carrier or first carrier, while the VPN user who is also a service provider is called a customer carrier or second carrier. This networking model is called carrier's carrier (CSC).

Figure 12 Model of Carrier's Carrier



Basic Concepts

- First carrier

The first carrier is also called a provider carrier, who provides MPLS VPN services for second carriers. In order to support second carriers to provide services for their users, the PE device of the first carrier must support CSC. The first carrier PE that provides services for second carriers is also called a CSC-PE.

- Second carrier

The second carrier is also called a customer carrier, who leases the MPLS L3VPN service from the first carrier in order to build its own intranet and then provide services for users. The second carrier CE connected to the first carrier is also called a CSC-CE.

- Internal route

The internal routes refer to the routes inside the network of the second carrier, namely, the intranet routes. The internal routes are used to guarantee the intercommunication of the second carrier's own network. Such routes must be jointly maintained by the first carrier PE and the second carrier.

- External route

Because the second carrier is a service provider, its network may be connected to multiple third-party networks. The route between the second carrier and the third-party network is called an external route. If the second carrier provides traditional IP services for users, the external routes include routes of user networks; if the second carrier is connected to the Internet, the external routes include Internet routes; if the second carrier provides MPLS VPN services for users, the external routes include user VPN routes.

Generally, there are tremendous external routes. To maintain good scalability, the first carrier will not maintain external routes, and external routes will be maintained independently by the second carrier.

- VPN tunnel

A VPN tunnel is an LSP tunnel established between VPN devices. In the CSC model, the LSP tunnel between second carrier devices is the VPN tunnel.

Working Principle

PE-CE Route and Label Distribution

To achieve good scalability, the number of routes to be maintained by the first carrier must be reduced. Therefore, the CSC model hands over external route maintenance to the second carrier, while the external traffic must use the VPN tunnel to traverse the first carrier. To support the CSC model, the first carrier PE must support VPN tunnels.

To support VPN tunnels, the first carrier PE (CSC-PE) and the second carrier CE (CSC-CE) must distribute the label binding information to each other. Depending on whether the CSC-PE and CSC-CE are in the same AS, the following routing protocols may be used to exchange internal routes and distribute labels for the internal routes:

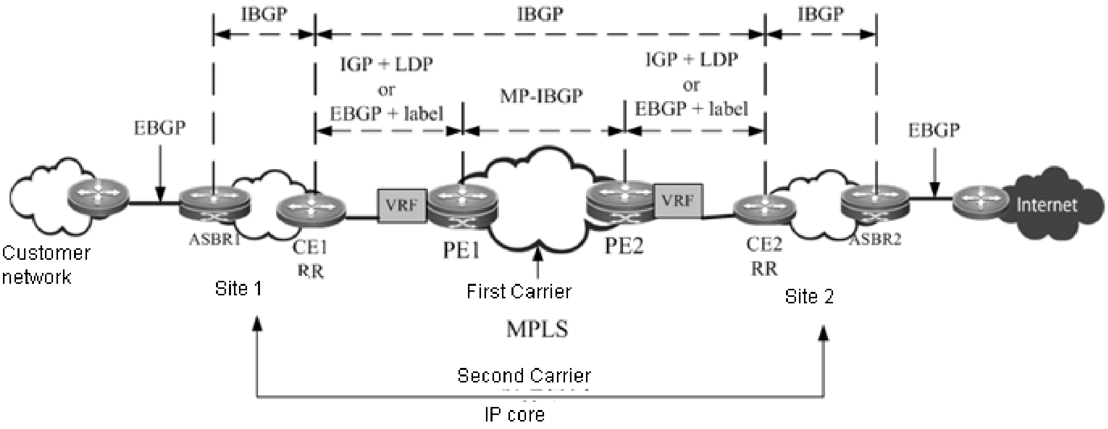
- If the CSC-PE and CSC-CE are in the same AS, IGP is generally used to exchange internal routes, and LDP is used to exchange label binding information.
- If the CSC-PE and CSC-CE are in different ASs, EBGP is generally used to exchange internal routes, and EBGP is enabled to perform label distribution for IPv4 routes and label distribution for internal routes.

Typical Application Scenarios

The second carrier can be an ordinary ISP or an MPLS service provider. Depending on the type of the second carrier network and the services provided by the second carrier for users, there are following typical application scenarios:

- Scenario I: IP core second ISP

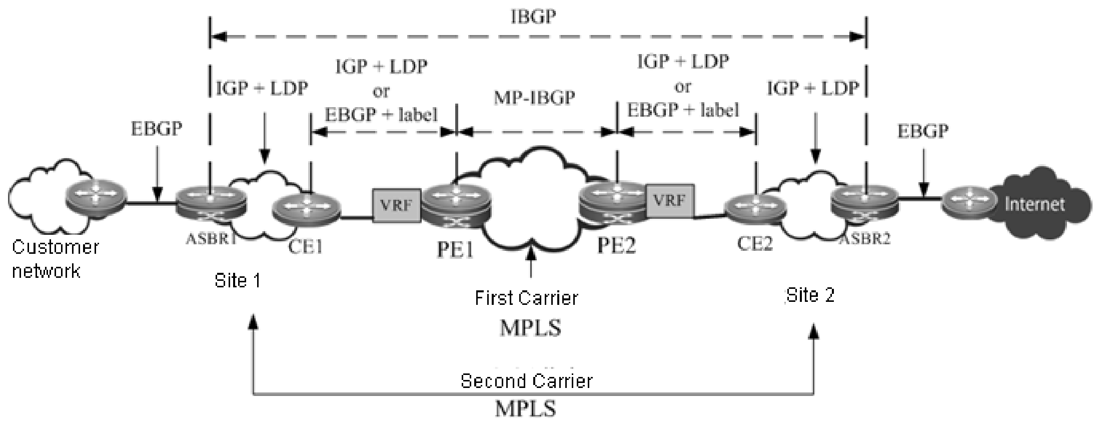
Figure 13 Scenario I: IP core second ISP



As shown in the figure, the second carrier is the IP core and provides network access services for users. IBGP neighbor relations are established between ASBR1, ASBR2, CE1 and CE2 to exchange external routes. CE1 and CE2 are RRs to reflect external routes between different sites. The Internet-access traffic of users flows from ASBR1 into the second carrier network and then flows out of the second carrier network from ASBR2. When the traffic flows from CE1 to CE2, the traffic is forwarded in the VPN LSP tunnel.

- Scenario II: MPLS core second ISP

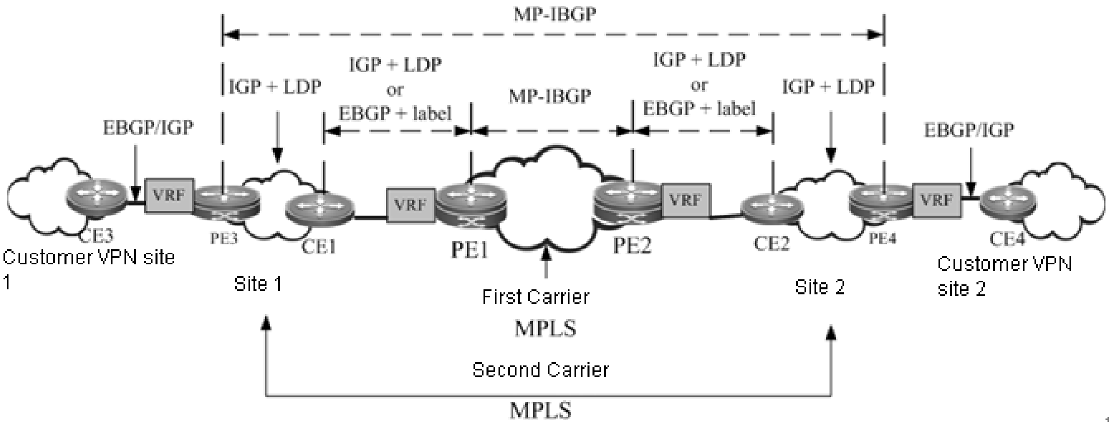
Figure 14 Scenario II: MPLS core second ISP



As shown in the figure, the second carrier is the MPLS core and provides network access services for users. An IBGP neighbor relation is established between ASBR1 and ASBR2 to exchange external routes. The Internet-access traffic of users flows from ASBR1 into the network of the second carrier and then flows out of the second carrier network from ASBR2. When the traffic flows from ASBR1 to ASBR2, the traffic is forwarded in the VPN LSP tunnel.

- Scenario III: MPLS core second VPN provider

Figure 15 Scenario III: MPLS core second VPN provider



As shown in the figure, the second carrier is the MPLS core and provides MPLS L3VPN services for users. An MP-IBGP neighbor relation is established between PE3 and PE4 to exchange user VPN routes. The VPN LSP between PE3 and PE4 acts as the external tunnel of a user VPN.

Configuration Steps

The CSC configuration includes:

- Configuring basic BGP/MPLS IP VPN functions for the first carrier
- Configuring the first carrier to enable CSC
- Configuring the second carrier
- Configuring user access for the second carrier

Configuring Basic BGP/MPLS IP VPN Functions for the First Carrier

The configuration of basic BGP/MPLS IP VPN functions includes:

42) Configuring an MPLS network

43) Configuring a VRF

44) Configuring an MP-IBGP neighbor

45) Configuring route exchange between PEs and CEs

- Configuring an MPLS network

The configuration in this section is similar to "Configuring an MPLS Network" in the "Configuring Basic BGP/MPLS IP VPN Functions" section.

- Configuring a VRF

The configuration in this section is similar to "Configuring a VPN Routing Instance" in the "Configuring Basic BGP/MPLS IP VPN Functions" section.

Caution

The CSC configuration requires "per-route" label allocation for each VRF. Therefore, you need to run the **alloc-label per-route** command in VRF configuration mode to select the label allocation mode.

- Configuring an MP-IBGP neighbor

The configuration in this section is similar to "Configuring PEs to Transmit VPN Routes" in the "Configuring Basic BGP/MPLS IP VPN Functions" section.

- Configuring route exchange between PEs and CEs

The configuration in this section is similar to "Configuring Route Exchange Between PEs and CEs" in the "Configuring Basic BGP/MPLS IP VPN Functions" section.

Caution

In Scenario I: In the network of the IP core second ISP, if the PE and CE use EBGP to exchange internal routes, because the external routes are exchanged using BGP and the CE is the route reflector, a route map needs to be configured for the PE and CE to filter external routes and avoid leaking external routes into the PE of the first carrier.

Configuring the First Carrier to enable CSC

Enable CSC on the first carrier PE. Depending on the protocol used for exchanging routes between a PE and a CE, the following two cases may apply:

46) The PE and CE use LDP to distribute labels.

47) The PE and CE use EBGP to distribute labels.

- The PE and CE use LDP to distribute labels.

If the PE and CE use IGP to exchange routes, run the following commands on the PE and CE respectively to configure the PE and CE to use LDP to distribute labels.

Use the following commands on the PE.

Command	Function

Command	Function
Ruijie(config)# **mpls router ldp** *vrf-name*	Enables LDP (PE) for the VRF.
Ruijie(config-mpls-router)# **ldp router-id interface** *interface-name* **force**	Configures the router ID of LDP.
Ruijie(config-mpls-router)# **advertise-labels for bgp-routes** [**acl** *acl-name*]	Configures LDP to distribute labels for BGP routes. LDP does not assign labels to BGP routes by default.
Ruijie(config-mpls-router)# **exit**	Exits LDP instance configuration mode.
Ruijie(config)# **interface** *interface-name*	Configures the interface connecting the CE.
Ruijie(config-if)# **label-switching**	Enables MPLS forwarding on the interface.
Ruijie(config-if)# **mpls ip**	Enables LDP on the interface.
Ruijie(config-if)# **ip ref**	☑ In case of a router, enables fast forwarding on the interface (not applicable to a switch).
Ruijie(config-if)# **end**	Exits interface configuration mode.
Ruijie# **show running**-config	Displays existing configuration information.
Ruijie# **show mpls ldp bindings vrf** *vrf-name*	Displays LDP label binding information under this VRF instance.

Use the following commands on the CE.

Command	Function
Ruijie(config)# **mpls router ldp**	Enables LDP (CE).
Ruijie(config-mpls-router)# **ldp router-id interface** *interface-name* **force**	Configures the router ID of LDP.
Ruijie(config-mpls-router)# **exit**	Exits LDP instance configuration mode.
Ruijie(config)# **interface** *interface-name*	Configures the interface connecting the PE.
Ruijie(config-if)# **label-switching**	Enables MPLS forwarding on the interface.
Ruijie(config-if)# **mpls ip**	Enables LDP on the interface.
Ruijie(config-if)# **end**	Exits interface configuration mode.
Ruijie# **show running-config**	Displays existing configuration information.
Ruijie(config-if)# **ip ref**	In case of a router, enables fast forwarding on the interface (not applicable to a switch).
Ruijie# **show mpls ldp bindings**	Displays LDP label binding information.

- The PE and CE use EBGP to distribute labels.

If the PE and CE use EBGP to exchange routes, run the following commands on the PE and CE respectively to configure the PE and CE to use EBGP to distribute labels.

Use the following commands on the PE.

Command	Function

Command	Function
Ruijie(config)# **interface** *interface-name*	Configures the interface connecting the CE.
Ruijie(config-if)# **ip ref**	☑ In case of a router, enables fast forwarding on the interface (not applicable to a switch).
Ruijie(config-if)# **label-switching**	Enables MPLS on the interface.
Ruijie(config-if)# **router bgp** *asn*	Enters BGP configuration mode.
Ruijie(config-router)# **address-family ipv4 vrf** *vrf-name*	Enters IPv4 address family configuration mode.
Ruijie(config-router-af)# **neighbor** {*peer-address* | *peer-group-name*} **send-label**	Enables BGP to carry labels for IP routes.
Ruijie(config-router-af)# **end**	Returns to privileged EXEC mode.
Ruijie# **show running-config**	Displays existing configuration information.
Ruijie# **show bgp vpnv4 unicast vrf** *vrf-name* labels	Displays BGP label information.

Use the following commands on the CE.

Command	Function
Ruijie(config)# **interface** *interface-name*	Configures the interface connecting the PE.
Ruijie(config-if)# **label-switching**	Enables MPLS on the interface.
Ruijie(config-if)# **ip ref**	☑ In case of a router, enables fast forwarding on the interface (not applicable to a switch).
Ruijie(config-if)# **router bgp** *asn*	Enters BGP configuration mode.
Ruijie(config-router)# **address-family ipv4**	Enters IPv4 address family configuration mode.
Ruijie(config-router-af)# **neighbor** {*peer-address* | *peer-group-name*} **send-label**	Enables BGP to carry labels for IP routes.
Ruijie(config-router-af)# **end**	Returns to privileged EXEC mode.
Ruijie# **show running-config**	Displays existing configuration information.
Ruijie# **show ip bgp labels**	Displays BGP label information.

Configuring the Second Carrier

Before the configuration, configure IGP for the second carrier network in order to guarantee the connectivity of the second carrier network. Depending on the application scenarios of the second carrier, different configuration schemes will be adopted:

- Scenario I: IP core second ISP
- Scenario II: MPLS core second ISP
- Scenario III: MPLS core second VPN provider

The Second Carrier Provides Internet Services Based on IP Core

In Scenario I, IBGP neighbor relations are established between ASBRs and CEs to exchange external routes. CEs are RRs to reflect external routes between different sites. The configuration task includes:

48) Configuring an intra-site BGP session

49) Configuring a BGP session between CSC-CEs of different sites

50) Configuring route map filtering

- Configuring an intra-site IBGP session

Use the following commands to configure an IBGP session between an intra-site ASBR and a CSC-CE, and configure the CSC-CE as an RR.

Command	**Function**
Ruijie(config)# **router bgp** *asn*	Configures a BGP router.
Ruijie(config-router)# **neighbor** {*peer-address* | *peer-group-name*} **remote-as** *asn*	Configures a BGP neighbor.
Ruijie(config-router)# **neighbor** {*peer-address* | *peer-group-name*} **route-reflector-client**	Configures the CSC-CE as the RR client.
Ruijie(config-router)# **neighbor** {*peer-address* | *peer-group-name*} **update-source** *interface-name*	Configures a BGP source address.
Ruijie(config-router)# **neighbor** {*peer-address* | *peer-group-name*} **next-hop-self**	For the ASBR, changes the next hop to the router itself when configuring the router to advertise BGP routes.

- Configuring an IBGP session between CSC-CEs of different sites

Use the following commands to set up a fully meshed IBGP session between CSC-CEs of different sites to exchange external routes of different sites.

Command	**Function**
Ruijie(config)# **router bgp** *asn*	Configures BGP.
Ruijie(config-router)# **neighbor** {*peer-address* | *peer-group-name*} **remote-as** *asn*	Configures a BGP neighbor.
Ruijie(config-router)# **neighbor** {*peer-address* | *peer-group-name*} **update-source** *interface-name*	Configures a BGP source address.
Ruijie(config-router)# **neighbor** {*peer-address* | *peer-group-name*} **route-reflector-client**	(Optional) Configures the CSC-CE of a peer site as the RR client.
Ruijie(config-router)# **exit**	Exits BGP configuration mode.
Ruijie(config)# **recursive-route lookup lsp**	Enables resolution of the next hop of a BGP route to an LSP tunnel.

- Configuring route map filtering

When BGP is used to exchange internal routes, because the CSC-CE is responsible for propagating both external routes and internal routes, you must guarantee that only the EBGP session between the CSC-CE and the CSC-PE can propagate internal routes, and that the IBGP session between CSC-CEs and between the CSC-CE and the ASBR can only propagate external routes; otherwise, routing loops or chaos may occur. To achieve this goal, you must run **neighbor route-map {in | out}** on the IBGP neighbor and EBGP neighbor to filter the corresponding routes, and the AS-path filtering rule is generally used. You can also use other rules.

Use the following commands to configure route map filtering.

Command	Function
Ruijie(config)# **ip as-path access-list** *access-list-number* {**permit** | **deny**} *regex*	Configures an AS-path ACL.
Ruijie(config)# **route-map** *route-map-name* {**permit** | **deny**} *sequence-number*	Configures a route map.
Ruijie(config-route-map)# **match as-path** *access-list-number*	Matches the AS-path ACL.
Ruijie(config-route-map)# **exit**	Returns to global configuration mode.
Ruijie(config)# **router bgp** *asn*	Configures a BGP router.
Ruijie(config-router)# **neighbor** {*peer-address* | *peer-group-name*} **route-map** *route-map-name* {**in** | **out**}	Applies the route map to a BGP neighbor.

The Second Carrier Provides Internet Services Based on MPLS

In Scenario II, the second carrier network is an MPLS core in which IBGP neighbor relations are established between ASBRs to exchange external routes. There is no need to propagate external routes via the CSC-CE. The configuration task includes:

51) Configuring an intra-site MPLS network

52) Configuring an inter-site IBGP session

■ Configuring an intra-site MPLS network

The configuration of the intra-site MPLS network of the second carrier is similar to "Configuring an MPLS Network" in the "Configuring Basic BGP/MPLS IP VPN Functions" section.

Note

You need to enable LDP on the CSC-CE in order to set up sessions with other intra-site devices to build an MPLS network. If the CSC-CE and CSC-PE use BGP to exchange routes, you must run **advertise-labels for bgp-routes** on the CSC-CE to enable LDP to distribute labels for BGP routes.

■ Configuring an IBGP session between ASBRs of different sites

Use the following commands to configure the BGP session between the local ASBR and the ASBR of a peer site in order to exchange external routes.

Command	Function
Ruijie(config)# **router bgp** *asn*	Configures BGP.
Ruijie(config-router)# **neighbor** {*peer-address* | *peer-group-name*} **remote-as** *asn*	Configures a BGP neighbor.
Ruijie(config-router)# **neighbor** {*peer-address* | *peer-group-name*} **update-source** *interface-name*	Configures a BGP source address.
Ruijie(config-router)# **neighbor** {*peer-address* | *peer-group-name*} **next-hop-self**	Changes the next hop to the router itself when configuring the ASBR router to advertise external routes.
Ruijie(config-router)# **exit**	Exits BGP configuration mode.
Ruijie(config)# recursive-route lookup lsp	Enables resolution of the next hop of a BGP route to an LSP tunnel.

Note

To reduce the configuration cost of a fully meshed IBGP session, you can configure the RR role inside the site. The intra-site ASBR can set up a BGP session with the RR, while an inter-site BGP session can only be set up between RRs.

The Second Carrier Provides VPN Services Based on MPLS Core

In Scenario III, the second carrier network is an MPLS core in which MP-IBGP neighbor relations are established between second carrier PEs to exchange user VPN routes. The configuration task includes:

53) Configuring an intra-site MPLS network

54) Configuring an MP-IBGP neighbor

■ Configuring an intra-site MPLS network

The configuration of the intra-site MPLS network of the second carrier is similar to "Configuring an MPLS Network" in the "Configuring Basic BGP/MPLS IP VPN Functions" section.

Note

You need to enable LDP on the CSC-CE in order to set up sessions with other intra-site devices to build MPLS network. If the CSC-CE and CSC-PE use BGP to exchange routes, you must run **advertise-labels for bgp-routes** on the CSC-CE to allow LDP to distribute labels for BGP routes.

■ Configuring PEs of each site to establish MP-IBGP neighbors

Configure PEs of each site to set up MP-IBGP sessions between intra-site PEs of the second carrier and between PEs of different sites in order to transmit VPN routes served by the second carrier. The configuration of the second carrier PE is similar to the PE configuration in the "Configuring Basic BGP/MPLS IP VPN Functions" section.

Note

To reduce the configuration cost of a fully meshed MP-IBGP session, you can configure the RR role inside the site. Intra-site PEs can set up an MP-IBGP session with the RR, while an inter-site MP-IBGP session can only be set up between RRs.

Configuring User Access for the Second Carrier

The configuration in this section relates to the services provided by the second carrier, and is irrelevant to the CSC model. If the second carrier provides IP services for users, see the "Configuring IP Routes" section. If the second carrier provides MPLS VPN services for users, see the "Configuring the MPLS VPN" section.

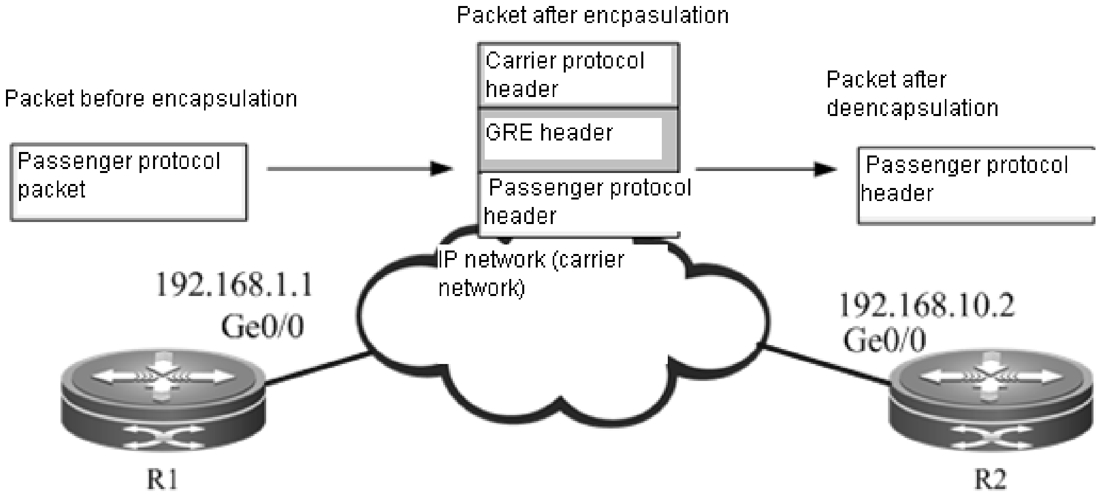
Configuring the MPLS VPN over GRE

☑ Currently, only router products of Ruijie support the MPLS VPN over GRE feature. This feature is not supported by Ruijie switch products.

Basic Concepts

The traditional MPLS VPN uses an LSP as the public tunnel, that is, VPN traffic flows from an upstream PE to a downstream PE by means of label switching. This requires the carrier's core network to fully support MPLS. For certain considerations or due to certain limitations, if the carrier's core network cannot fully support MPLS, the MPLS VPN over GRE can provide a mechanism to allow the carrier to use a GRE tunnel as a hop on the LSP tunnel to guarantee the integrity of the public LSP.

Figure 16 GRE tunnel


Packet after encpasulation
Carrier protocol header
GRE header
Passenger protocol header
Packet before encapsulation
Passenger protocol packet
Packet after deencapsulation
Passenger protocol header
IP network (carrier network)
192.168.1.1
Ge0/0
192.168.10.2
Ge0/0
R1
R2

- GRE tunnel

GRE provides a mechanism to encapsulate the packets of one protocol (passenger protocol) into another protocol (carrier protocol). The encapsulated packets consist of: carrier protocol header, GRE header and original passenger protocol header. After being encapsulated by the carrier protocol, the passenger protocol packets can be forwarded in the carrier network. After the encapsulated packets reach the destination address of the carrier protocol, the destination device will decapsulate the packets and then forward the packets according to the inner-layer passenger protocol used by packets. Such an encapsulation technology allows passenger protocol packets to traverse a heterogeneous carrier network and reach the destination device. It is a tunnel encapsulation technology.

- Passenger protocol

The passenger protocol is the protocol being encapsulated during the process of GRE encapsulation. In the application scenario of MPLS VPN over GRE, the passenger protocol refers to packets carrying MPLS labels.

- Carrier protocol

The carrier protocol is the protocol used to encapsulate the passenger protocol during the process of GRE encapsulation. In the application scenario of MPLS VPN over GRE, the carrier protocol is generally IPv4.

- Source address and destination address

While encapsulating the passenger protocol, you need to know the source address and destination address of the carrier protocol, so that the encapsulated packets can be forwarded on the carrier network. The abovementioned source address and destination address are the source address and destination address of a GRE tunnel.

- Tunnel endpoint

When packets are transported on the tunnel, one device carries out carrier protocol encapsulation and another device carries out decapsulation. The passenger protocol information can only be known and processed by these two devices, while other carrier network devices between the two devices are unaware of the existence of the passenger protocol. These two devices are the endpoints of the GRE tunnel.

Working Principle

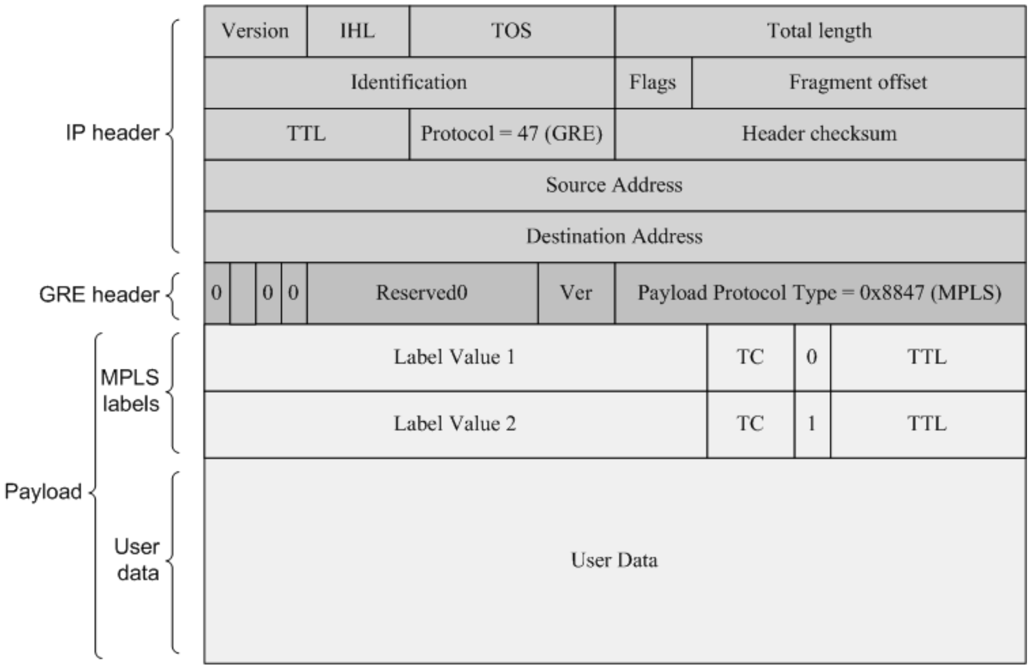
In the traditional MPLS VPN, private-network traffic carrying an inner-layer VPN label and an outer-layer public-network label reaches the peer PE by means of label switching. When a non-MPLS network exists in the backbone network, the LSP will become discontinuous. The GRE tunnel can help MPLS packets traverse a non-MPLS domain and realize a continuous LSP.

The GRE tunnel is a tunneling mechanism in an IP network and support GRE with MPLS as the passenger protocol, so that two devices on both sides of the IP network can exchange MPLS packets. Considering the GRE tunnel as a point-to-point logical link, devices at both ends of the tunnel directly establish IGP neighbor relations and LDP neighbor relations on this link to distribute routes and labels for the LSP, while the GRE tunnel becomes one hop of the LSP.

MPLS as a Passenger Protocol

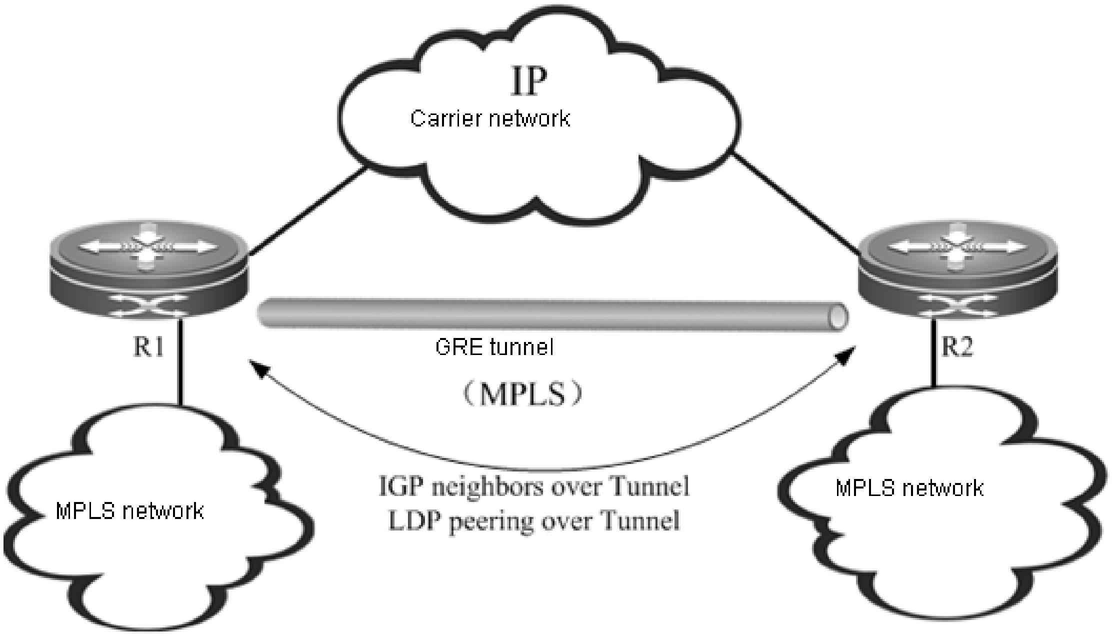
MPLS is used as the GRE tunnel of the passenger protocol so that two devices interconnected through a non-MPLS network can forward MPLS packets to each other. After a label operation at one end of the tunnel, MPLS packets are GRE-encapsulated and then transported over the carrier network to the other end of the tunnel; label switching is then carried out after packet decapsulation at the other end of the tunnel. The following figure shows the format of encapsulated packets with IPv4 being the carrier protocol and MPLS being the passenger protocol.

Figure 17 MPLS as the passenger protocol



GRE Tunnel as a Point-To-Point Link

Figure 18 GRE tunnel link



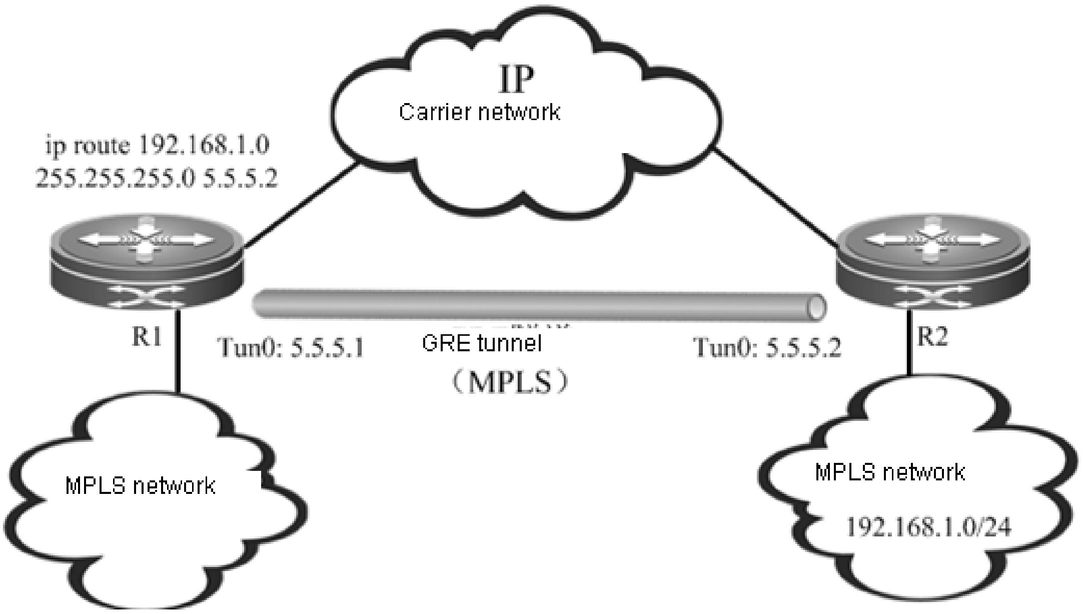
As shown in the figure, R1 and R2 are connected to an MPLS network respectively, while both routers are interconnected via a carrier network (IP). The GRE tunnel allows both endpoints (R1 and R2) to use the carrier network (IP) to transmit MPLS packets, so that two separated MPLS networks can be connected. The GRE tunnel is the point-to-point logical link between R1 and R2. It bypasses the carrier network (IP) and becomes one part of the MPLS network, so that the MPLS networks at both ends of the tunnel can maintain continuity. Considering the GRE tunnel as a point-to-point link, the IGP protocol can run on the link, while LDP can also distribute labels between R1 and R2.

Introduction of Tunnel Traffic

In either the carrier network (IP) or the MPLS network, traffic forwarding is driven by routers. Therefore, a dynamic routing protocol needs to be run in the carrier network (IP) and MPLS network. There are two possible schemes: single routing instance and dual routing instances.

- Single routing instance

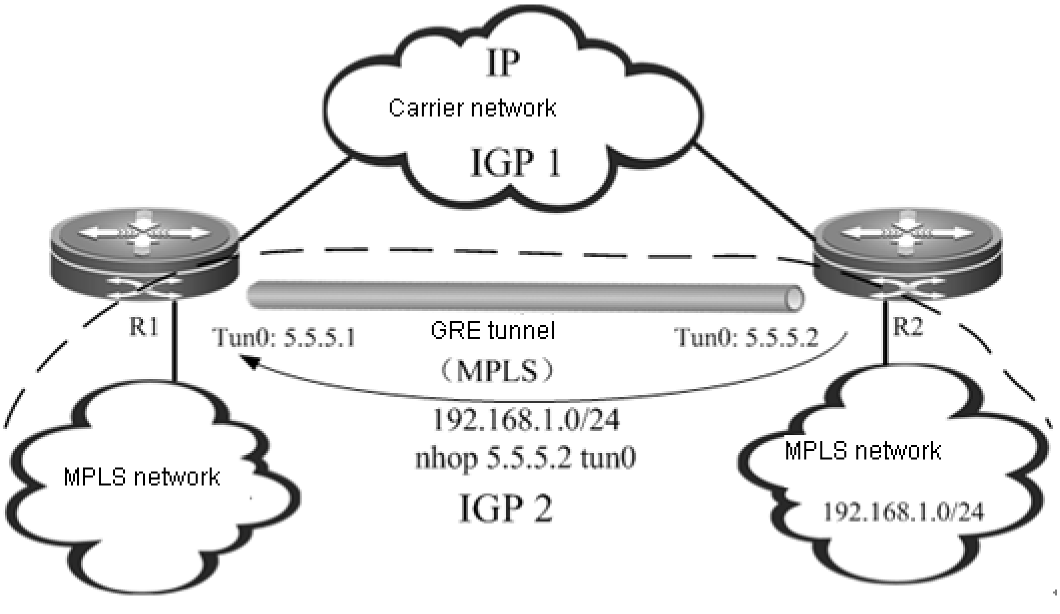
Figure 19 Single routing instance



In this scheme, the MPLS network and carrier network (IP) are in the same routing instance, and the entire network is in the plane form, as shown in Figure 15. By default, because the metric value of the GRE tunnel is far greater than that of an ordinary link, no traffic will be introduced into the GRE tunnel (which means that the GRE tunnel is not the next-hop egress interface of any route). Therefore, you must configure static routes in order to introduce MPLS traffic into the GRE tunnel. The static routes must be configured in this scheme, and the number of static routes depends on the number of route prefixes to be introduced into the GRE tunnel. The scalability is not satisfactory.

- Dual routing instances

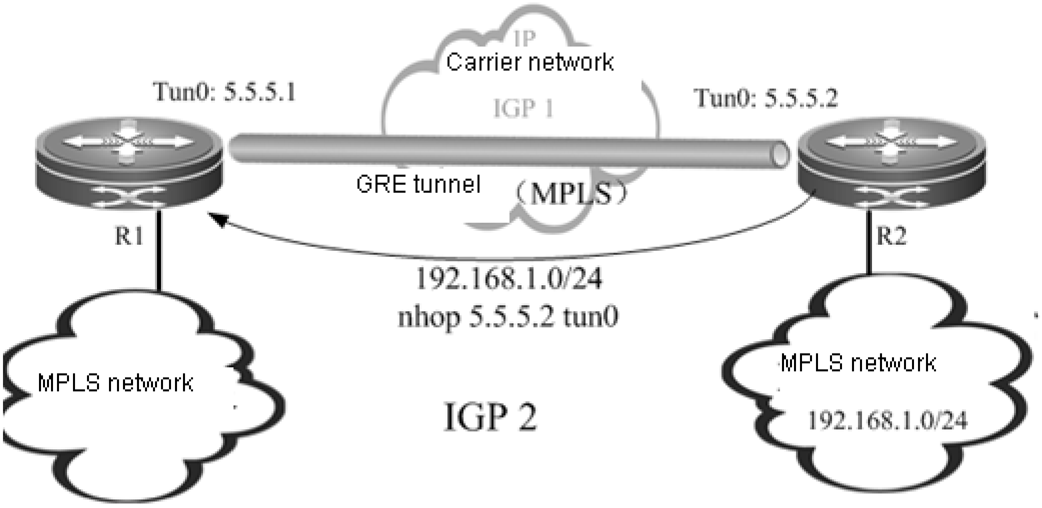
Figure 20 Dual routing instances



In this scheme, there are two different routing instances on each endpoint device of the GRE tunnel, as shown in Figure 16. One routing instance participates in route exchange in the carrier network (IP), while the other routing instance will participate in the route exchange in the MPLS network (including the GRE tunnel link). At this time, R1 learns the route to the remote MPLS network through the GRE tunnel, with the egress interface being the GRE tunnel. The traffic can be introduced into the GRE tunnel without configuring any static route.

Dual routing instances are actually dividing the network into different layers. As the upper-layer network, the MPLS network (including the GRE tunnel) acts as the backbone network running consistent IGP instances, supporting MPLS and providing MPLS VPN services. As the bottom-layer network, the carrier network (IP) is the local network between R1 and R2 and runs independent IGP instances. If the GRE tunnel is the "layer-3 interface" between R1 and R2, the IP network and the IGP instance between R1 and R2 will be the "layer-2 network" and "layer-2 link protocol" between R1 and R2, as they guarantee the link state of the GRE tunnel. The relation can be indicated in Figure 17.

Figure 21 Dual IGP instances

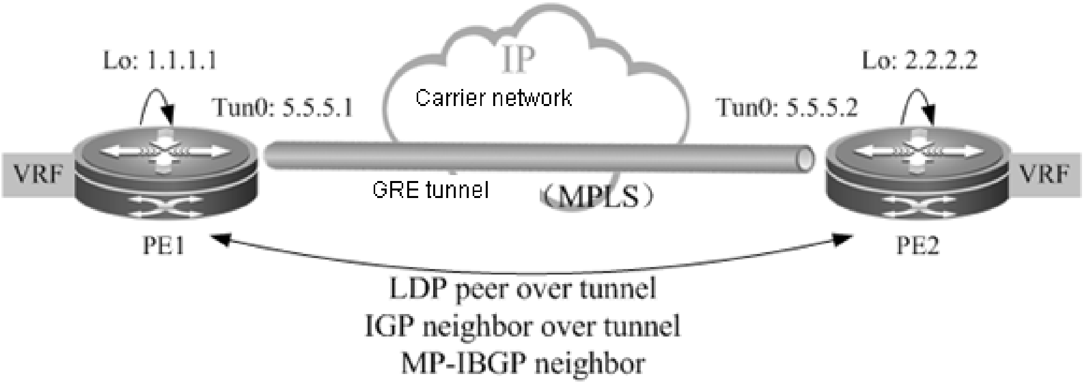


The scheme of dual routing instances divides the network into different layers and boasts better scalability. The following example is mainly based on this scheme.

Typical Applications

- Establishing a GRE tunnel between PEs

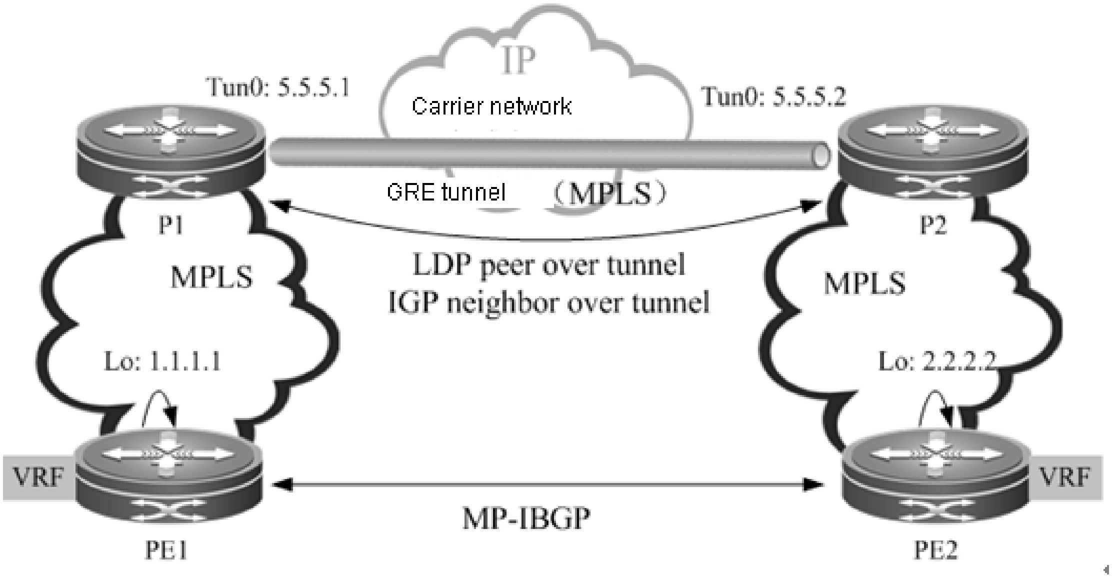
Figure 22 Scenario I: PE-PE



As shown in Figure 18, the core network between PEs is completely an IP network. The GRE tunnel is established between two PEs, and the LSP between PE1 and PE2 has only one hop.

- Establishing a GRE tunnel between Ps

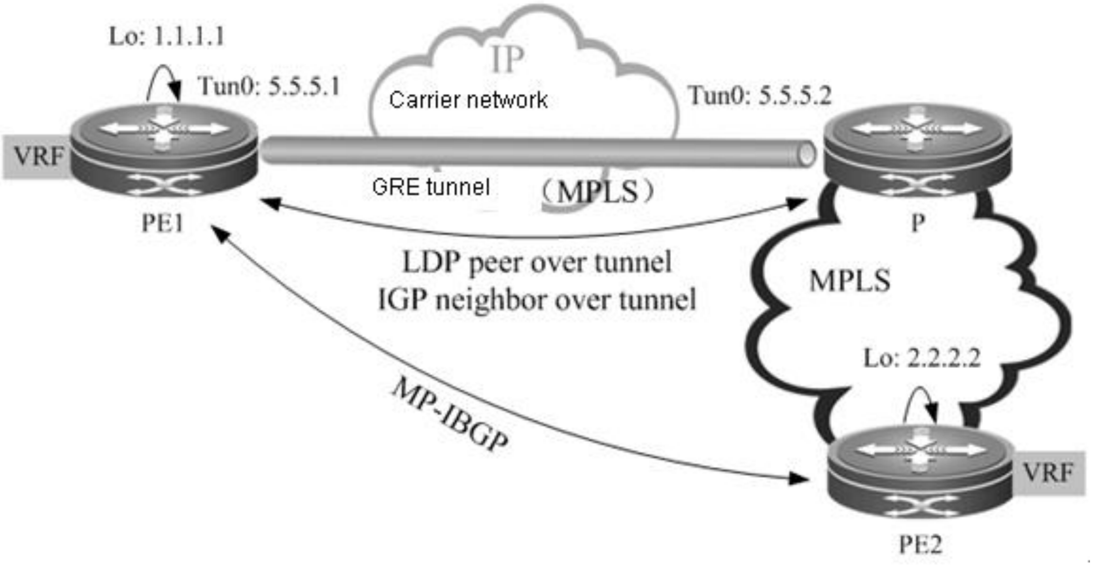
Figure 23 Scenario II: P-P



As shown in Figure 19, PE1 and PE2 are in two MPLS domains. P1 and P2 are interconnected through an IP network. The GRE tunnel is established between P1 and P2. The public LSP between PE1 and PE2 goes through P1 and P2, and the GRE tunnel between P1 and P2 is one hop of the LSP.

- Establishing a GRE tunnel between P and PE

Figure 24 Scenario III: P-PE



As shown in Figure 20, the network between PE1 and P does not support MPLS. The LSP is connected by establishing a GRE tunnel between PE1 and P.

Protocol Specification

- RFC 4023: Encapsulating MPLS in IP or GRE.

- RFC 4797: Use of Provider Edge to Provider Edge (PE-PE) Generic Routing Encapsulation (GRE) or IP in BGP/MPLS IP Virtual Private Networks.

Configuration Steps

The configuration of the MPLS VPN over GRE includes:

- Creating a tunnel interface
- Configuring an IGP route
- Configuring an MPLS network
- Configuring an MPLS VPN

Creating a Tunnel

Use the following commands to create a GRE tunnel (interface).

Command	Function
Ruijie(config)# **interface tunnel** *tunnel-id*	Creates a tunnel interface.
Ruijie(config)# **tunnel mode gre ip**	Configures the tunnel as a GRE in IP tunnel.
Ruijie(config-if)# **ip address** *ip-address address-mask*	Configures an address for the tunnel interface.
Ruijie(config-if)# **tunnel source** {*ip-address* | *interface-name*}	Configures the source address of the GRE tunnel.
Ruijie(config-if)# **tunnel destination** *ip-address*	Configures the destination address of the GRE tunnel.
Ruijie(config-if)# **no shutdown**	Enables the interface.

Configuring a Route to Introduce Traffic into the Tunnel

There are two ways to introduce traffic into the tunnel:

55) Configuring IGP

56) Configuring a static route

- Configuring IGP

Generally, multiple OSPF processes are used to create different routing instances. One OSPF process learns the route to the destination address of the tunnel, so that the tunnel interface is up if the route is reachable. Another OSPF process runs OSPF on the GRE tunnel to set up a session in order to learn the route to the destination address of the PE. For the configuration steps of multiple OSPF processes, see the "Configuring the Unicast Routing Protocol" section.

- Configuring a static route

Use the following command to configure a static route directly: Configure the tunnel interface as the egress of the host route to the specified PE address.

Command	Function
Ruijie(config)# **ip route** ip-address address-mask **tunnel** tunnel-id	Configures a static route.

Caution

If a static route is used to introduce traffic into the tunnel, the destination address of the tunnel cannot be the route prefix of the static route, that is, the address of the specified PE and the destination address of the tunnel must be different. This is because the state of the tunnel interface depends on the route to the destination address of the tunnel, while the static route will cause the route to the destination address to rely on the state of the tunnel interface, thus leading to the state flapping of the tunnel interface.

Configuring the Tunnel Interface to Enable MPLS

Use the following commands to enable LDP on the tunnel interface and enable MPLS forwarding.

Command	Function
Ruijie(config)# **mpls ip**	Enables MPLS globally. ☑ This command is not available on a switch chip.
Ruijie(config)# **mpls router ldp**	Enables LDP globally.
Ruijie(config-mpls-router)# **ldp router-id interface** *interface-name*	Configures the router ID of LDP.
Ruijie(config-mpls-router)# **exit**	Exits LDP configuration mode.
Ruijie(config)# **interface tunnel** *tunnel-id*	Enters tunnel interface configuration mode.
Ruijie(config-if)# **ip ref**	☑ In case of a router, enables fast forwarding (not applicable to a switch).
Ruijie(config-if)# **mpls ip**	Enables LDP on the interface.
Ruijie(config-if)# **label-switching**	Enables MPLS forwarding on the interface.
Ruijie(config-if)# **exit**	Exits interface configuration mode.

☑ Currently, only router products support the MPLS VPN over GRE-in-IPv4 tunnel. This feature is not supported by switch products.

Configuring the MPLS VPN

The configuration of the MPLS VPN includes:

- Configuring a VRF
- Configuring an MP-IBGP
- Configuring route exchange between PEs and CEs

For details, see the "Configuring Basic BGP/MPLS IP VPN Functions" section.

Configuring the OSPF VPN Extension

Understanding the L3VPN OSPF VPN Extension

PE-CE OSPF Feature

OSPF is a widely used IGP protocol. In most of the existing application schemes, VPN users generally select OSPF as the interior routing protocol. If OSPF is used between a PE and a CE, you do not need to run other routing protocols, thus simplifying CE configuration and management.

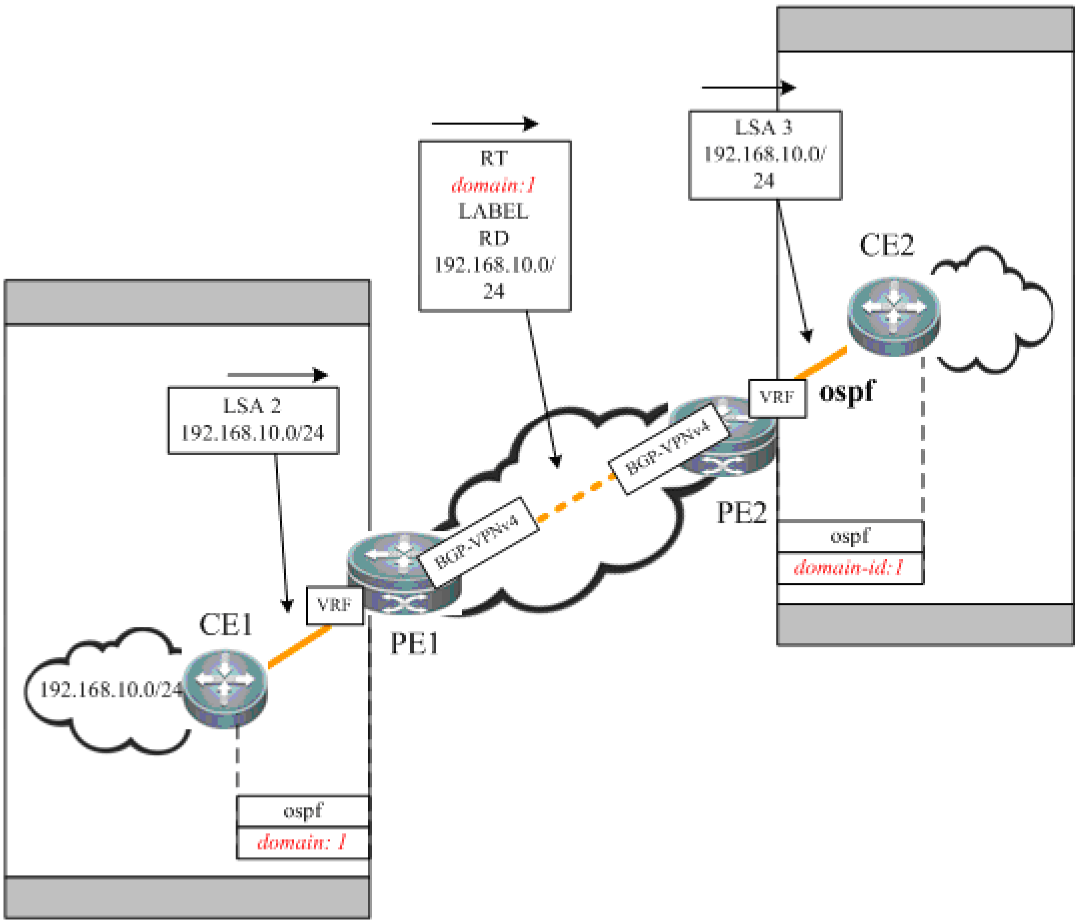
The PE-CE OSPF feature is described from the following four aspects.

- Domain ID

A domain ID refers to the OSPF domain to which the route belongs. When the CE has learned an OSPF route from an intra-VPN site, this route will be advertised to a PE in type-1, type-2 or type-3 link state advertisements (LSAs) and redistributed to BGP to form a VPN route. Meanwhile, the domain ID will also be redistributed to BGP together with the route and advertised as the extended community attribute in the VPN route. When other PEs receive this VPN route and redistribute it to the VRF OSPF instance, the domain ID will also be redistributed to the corresponding VRF OSPF instance together with this route. If the VRF OSPF instance confirms that the domain ID contained in the route is the same as the domain ID of this VRF OSPF instance, the route will be advertised to the CE as an internal route. Contrarily, if the VRF OSPF instance confirms the domain ID contained in the route is different from the domain ID of this VRF OSPF instance, the route will be advertised to the CE as an external route.

As shown in the following figure, for a route that belongs to the same OSPF domain, CE1 advertises the route to PE1 in a type-2 LSA and then a VPN route is formed and advertised to PE2. PE2 receives this route and redistributes it to the VRF OSPF instance. Because the VRF OSPF instance shares the same domain ID with this VPN route, this site will eventually be advertised to VPN sites in the form of an internal route.

Figure 25

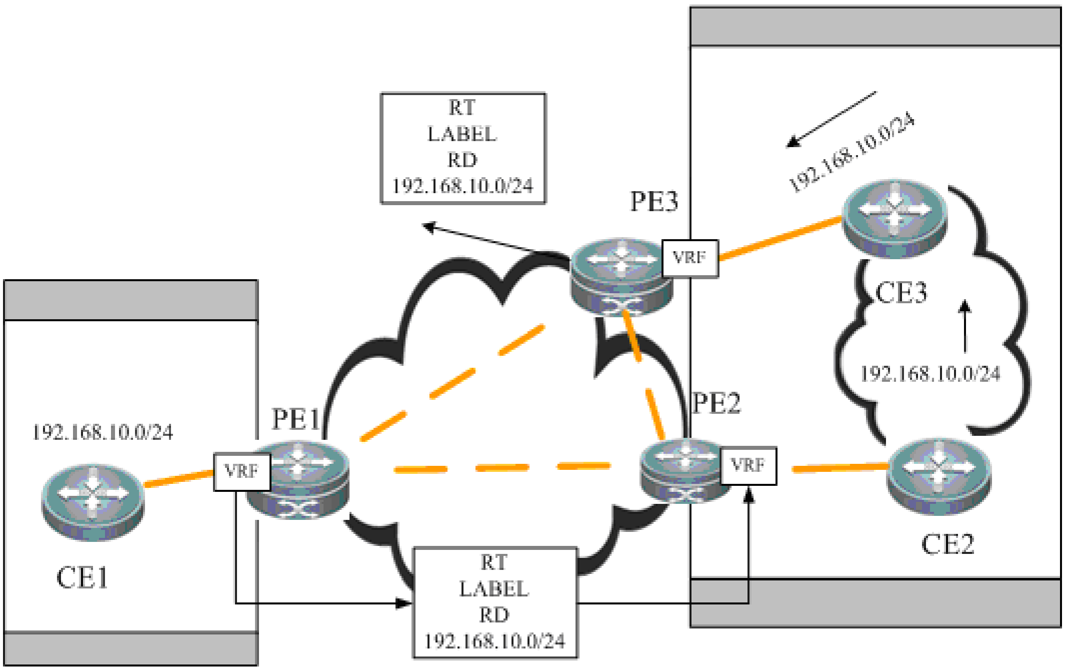


- DN bit

A DN bit is a loop detection technique when OSPF is run between the PE and the CE. In a certain scenario, running OSPF between the PE and the CE may cause loops. For example, when multiple PEs are connected to one VPN site, if one PE advertises the VPN route learned to the VPN site, and the route is further advertised to another PE by running OSPF inside the VPN site and then propagated, a routing loop may take place.

As shown in the following figure: The route from 192.168.10.0/24 is advertised by PE1 to PE2 and PE3. CE2 advertises the route to CE3 through OSPF. The route is then advertised to PE3 and redistributed to the BGP protocol of PE3. PE3 selects the route redistributed by OSPF and converts this route into a VPN-IPV4 route before advertising the route, thus causing a routing loop.

Figure 26



To avoid such potential loop, when the PE advertises a type-3, type-5 or type-7 LSA to the CE, it will set a DN bit in an optional field of the LSA. When other PEs receive any LSA containing a DN bit in the optional field, the OSPF protocol on the PE does not allow this LSA to participate in OSPF route computation.

- VPN route tag

A VPN route tag is another loop detection technique. When OSPF is run between the PE and the CE, the corresponding VRF OSPF instance on the PE will by default have a route tag called "VPN route tag". The VRF OSPF instance on the PE imports the VPN route and converts the route into a type-5 or type-7 LSA. When LSA is advertised to CE, this LSA will carry a VPN route tag. In the circumstance in which one VPN site is connected with multiple PEs, if a PE receives a type-5 or type-7 LSA that carries a VPN route tag and this VPN route tag is the same as the that of the OSPF instance, this LSA will not participate in OSPF route calculation.

- PE-CE inter-area deployment

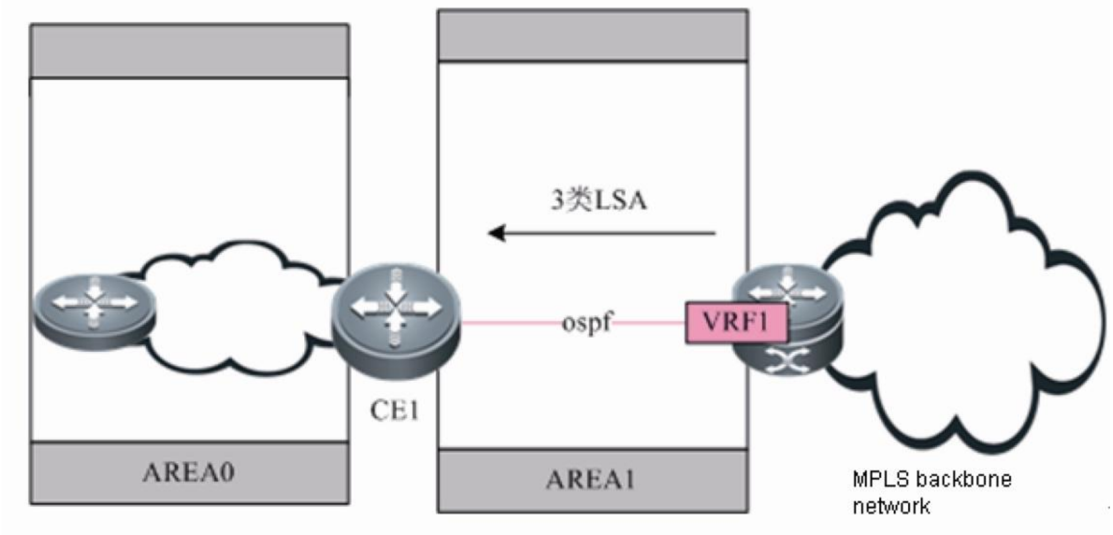
Under normal circumstances, the link between the PE and the CE can be in any OSPF area. However, if the link between the PE and the CE falls into a non-zero area, the PE is an area border router (ABR) to the OSPF area where the CE is located. This may cause some problems because the OSPF protocol acting as an ABR device has the following features:

57) The ABR only calculates the type-3 LSA in the backbone area.

58) The ABR only forwards the type-3 LSA in the backbone area to a non-backbone area.

As shown in the following figure, if the link between the PE and the CE is in a non-zero area, the PE will redistribute the VPNv4 route advertised by MP-BGP to OSPF and restore it to a type-3 LSA to be advertised to CE1. CE1 will not calculate the non-backbone area LSAs. Therefore, these LSAs will not be advertised to routers in Area 0, and intra-VPN sites may fail to learn the route to other sites. Therefore, pay special attention when deploying a non-zero area between the PE and the CE.

Figure 27

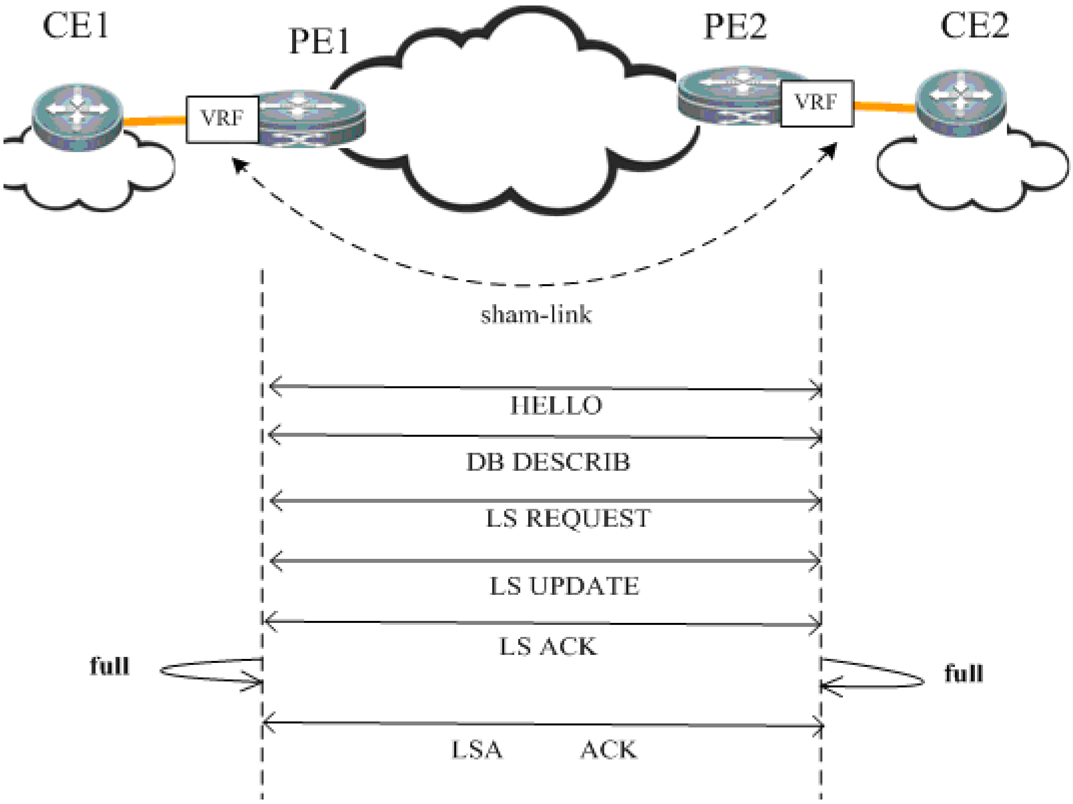


Generally, in L3VPN applications, if OSPF is run between the PE and the CE to exchange VPN routes, it is suggested not to deploy the backbone area at an intra-VPN site. In practice, if intra-VPN routers other than the PE sites also fall into the backbone area, there must be at least one router at this intra-VPN site to connect to the PE, and the link between the CE and the PE must belong to Area 0, so that intra-area routes and external routes can be propagated between the PE and the VPN site.

Sham Link

A sham link is not a real link. It refers to a "virtual link" established between the VRFs of two PEs. The sham link is the same as the normal OSPF link. With its own OSPF interface, it can send OSPF protocol packets, establish neighbors, and send LSAs. When LSAs are flooded over the sham link, all OSPF route types will not be changed, as shown in the following figure.

Figure 28



The purpose of establishing a sham link between VRF OSPF instances on different PEs is as follows:

- The approach of using MP-IBGP to carry a private route will only propagate the route, and the restoration after reaching the peer PE is only to import the original OSPF route information as far as possible, during which the OSPF topology information cannot be truly communicated. By establishing an OSPF link through a sham link, all OSPF instances inside each site can be truly connected and work out complete topology information.
- Different sites in the same VPN exchange information via the MPLS backbone network, but a link is established between VPN sites so that VPN sites can still communicate via this link when the MPLS backbone network fails. This link is called the "backdoor link". If two sites of VPN users fall into the same OSPF area and there is a "backdoor link" connecting these two sites, routes will be exchanged via both the MPLS backbone network and the "backdoor link". Because the routes exchanged via the MPLS backbone network are inter-area routes and the routes exchanged via the "backdoor link" are intra-area routes, and the intra-area routes are apparently superior to inter-area routes, the route forwarding between two sites will hence use the backdoor link. This goes against the purpose of establishing the "backdoor link". Therefore, the sham link must be used in such applications.

Protocol Specification

- RFC 4576 and RFC 4577 specify the mechanism to realize L3VPN OSPF.

Default Configuration

Function	Default Setting
domain-tag	AS number of local BGP
domain-id	Default value: NULL; default type: 0x0005

Function	Default Setting
capability vrf-lite	The PE-CE OSPF feature is supported by the VRF OSPF instance by default.
extcommunity-type	In the OSPF instance extended community attribute carried by BGP, the default type of route-id is 0x0107, and the default type of route-type is 0x0306.

Configuring a Domain ID (Optional)

The domain ID is used to indicate the domain to which the OSPF instance belongs. Generally, all VRF OSPF instances belonging to the same VPN must use the same domain ID.

Use the following commands to configure the domain ID.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **router ospf** *ospf-id* **vrf** *vrf-name*	Creates an OSPF instance and enters OSPF configuration mode.
Ruijie(config-router)# **domain-id** {*ip-address* [secondary] | **null | type** {0005|0105|0205|8005} **value** *hex-value* **[secondary]**}	(Optional) Configures the domain ID of the OSPF instance. The OSPF domain ID is 0 by default.
Ruijie(config-router)# **show running-config**	Displays existing configuration information.

☑ Supported by RSR20 series running 10.4(3) or later versions

☑ Supported by RSR30 series running 10.4(3) or later versions

☑ Supported by RSR50 series running 10.4(3) or later versions

☑ Supported by RSR50E series running 10.4(3) or later versions

Note

This command is only applicable to an OSPF instance associated with a VRF.

A VRF OSPF instance can be configured with multiple domain IDs, but there is only one primary domain ID. Others are secondary domain IDs. The only primary domain ID is configured with the **domain-id** *value* command, while multiple secondary domain IDs are configured with the **domain-id** *value* **secondary** command. OSPF routes are advertised when converted into VPN routes, and VPN routes only contain the primary domain ID.

You can use the **domain-id** *ip-address* command or the **domain-id type {0005|0105|0205|8005}** *value* command to configure the primary and secondary domain IDs.

Different VRF OSPF instances can have the same domain ID. However, VRF OSPF instances in the same VPN must be configured with the same domain ID in order to guarantee the correctness of route advertisements.

# Configure the primary domain ID and secondary domain ID of the VRF OSPF protocol to 4.4.4.4 and 5.5.5.5 respectively.

```
Ruijie# configure terminal
```

```
Ruijie(config)# router ospf 10 vrf vrf1
Ruijie(config-router)# domain-id 4.4.4.4
Ruijie(config-router)# domain-id 5.5.5.5 secondary
Ruijie(config-router)# domain-id type 0005 value 010101010101 secondary
```

Configuring a VPN Route Tag (Optional)

Use the following commands to configure a VPN route tag.

Command	**Function**
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# r**outer ospf** *ospf-id* **vrf** *vrf-name*	Creates an OSPF instance and enters OSPF configuration mode.
Ruijie(config-router)# **domain-tag** *value*	(Optional) Configures a VPN route tag (1 to 4294967295) for the OSPF instance.
Ruijie(config-router)# **show running-config**	Displays existing configuration information.

☑ Supported by RSR20 series running 10.4(3) or later versions

☑ Supported by RSR30 series running 10.4(3) or later versions

☑ Supported by RSR50 series running 10.4(3) or later versions

☑ Supported by RSR50E series running 10.4(3) or later versions

Note

This command is only applicable to an OSPF instance associated with a VRF.

If the domain tag of a VRF is not configured manually, the default value is the AS number of the local BGP protocol.

In an L3VPN, if one VPN site is connected with multiple PEs, the VPN route learned by a PE through MP-BGP will be advertised to a VPN site in a type-5 or type-7 LSA. The route may also be learned by other PEs connecting to this VPN site and then advertised, hence causing a loop. To avoid such loop, the same VPN route tag must be configured on the PE for VRF OSPF instances connected to the same VPN site.

When a VRF OSPF instance sends a type-5 or type-7 LSA to the VPN site, this LSA will also carry the VPN route tag. When other PEs receive such type-5 or type-7 LSA containing the VPN route tag, if such route tag is the same as the route tag of the corresponding OSPF instance, this LSA will not participate in OSPF route computation.

Generally, OSPF instances associated with the same VPN must be configured with the same VPN route tag.

# Configure the domain tag of the VRF OSPF protocol to 10.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 10 vrf vrf1
Ruijie(config-router)# domain-tag 10
```

Configuring a Sham Link (Optional)

The sham link is mainly used in the scenario where there is a backdoor link between VPN sites. If you still expect to transmit VPN data via the MPLS backbone network, you can establish a sham link between the VRF OSPF instances of two PEs. Both instances can establish OSPF neighbors through this sham link and distribute LSA packets over the sham link.

Use the following commands to configure a sham link.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **router ospf** *ospf-id* **vrf** *vrf-name*	Creates an OSPF instance and enters OSPF configuration mode.
Ruijie(config-router)# **area area-id sham-link** *source-address destination-address* [**cost** *number*] [**dead-interval** *seconds*] [**hello-interval** *seconds*] [**retransmit-interval** *seconds*] [**transmit-delay** *seconds*] [**authentication** [**message-digest** | **null**]] [[**authentication-key [0|7]** *key*] | [**message-digest-key** *key-id* **md5 [0|7**] *key*]]	Configures the area ID, source address and destination address of the sham link.
Ruijie(config-router)# **show running-config**	Displays existing configuration information.

☑ Supported by RSR20 series running 10.4(3) or later versions

☑ Supported by RSR30 series running 10.4(3) or later versions

☑ Supported by RSR50 series running 10.4(3) or later versions

☑ Supported by RSR50E series running 10.4(3) or later versions

Note

The sham link must be configured on two PEs that intend to establish the sham link. The sham link cannot be established if only one PE is configured.

The following conditions must be met in order to establish the sham link between two PEs:

1.The area-id of the sham link configured on two PEs must be identical.

2.The source address and destination address of the sham link configured on one PE must correspond to the destination address and source address of the sham link configured on another PE.

3.The source address and destination address used to establish the sham link on the PE must be a 32-bit loopback address bound to a VRF.

Because OSPF routes advertised by a sham link do not carry VPN labels, the routes cannot be used to forward packets. Packets are forwarded through BGP VPNv4 routes. Therefore, there must be VPNv4 routes corresponding to the OSPF routes learned from the sham link.

The source address for a sham link is advertised in BGP VPNv4 routes, but not calculated in OSPF instances.

# Configure a sham link for a VRF OSPF instance, with the area ID being 0, the source address being 1.1.1.1, and the destination address being 2.2.2.2.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 10 vrf vrf1
Ruijie(config-router)# area 0 sham-link 1.1.1.1 2.2.2.2
```

Configuring Capability vrf-lite (Optional)

The PE-CE OSPF feature of the VRF OSPF instance includes LSA conversion according to the domain ID, DN bit, and VPN route tag. In certain circumstances, if you do not expect the VRF OSPF instance to support the PE-CE OSPF feature, you can use the **capability vrf-lite** command to disable the feature.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **router ospf** *ospf-id* **vrf** *vrf-name*	Creates an OSPF instance and enters OSPF configuration mode.
Ruijie(config-router)# **capability vrf-lite**	(Optional) Disables the PE-CE OSPF feature of the VRF OSPF instance.
Ruijie(config-router)# **show running-config**	Displays existing configuration information.

☑ Supported by RSR20 series running 10.4(3) or later versions

☑ Supported by RSR30 series running 10.4(3) or later versions

☑ Supported by RSR50 series running 10.4(3) or later versions

☑ Supported by RSR50E series running 10.4(3) or later versions

☑ Supported by 3760E series running 10.4(3) or later versions

☑ Supported by 5750 series running 10.4(3) or later versions

☑ Supported by 5760E series running 10.4(3) or later versions

Note

This command is only applicable to an OSPF instance associated with a VRF.

In certain scenarios, you may expect to disable the loop detection function of the VRF OSPF instance. For example: A VPN user uses an MCE device to exchange VPN routes with a PE. If the MCE and PE exchange VPN routes via OSPF, to allow the VPN site to learn the routes of other VPN sites, you must use the **capability vrf-lite** command on the MCE device to disable the loop detection function of the VRF OSPF instance.

# Disable the loop detection function of the VRF OSPF instance.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 10 vrf vrf1
Ruijie(config-router)# capability vrf-lite
```

Configuring extcommunity-type (Optional)

When an OSPF route is redistributed to BGP to form a VPN route, the extended community attributes of the OSPF route are also attached, including router-id and route-type. By default, the type of the extended community attribute of router-id is 0x0107, and that of route-type is 0x0306.

You can use the following commands to manually configure the extended community attributes of rouer-id and route-type.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **router ospf** *ospf-id* **vrf** *vrf-name*	Creates an OSPF instance and enters OSPF configuration mode.
Ruijie(config-router)# **extcommunity-type {router-id {0107|8001} | route-type {0306|8000}}**	(Optional) Configures the extended community attributes of OSPF router-id and route-type.
Ruijie(config-router)# **show running-config**	Displays existing configuration information.

☑ Supported by RSR20 series running 10.4(3) or later versions

☑ Supported by RSR30 series running 10.4(3) or later versions

☑ Supported by RSR50 series running 10.4(3) or later versions

☑ Supported by RSR50E series running 10.4(3) or later versions

Note

This command is only applicable to an OSPF instance associated with a VRF.

The type configuration of router-id provides good compatibility with multiple manufacturers. For example, some manufacturers support only the router-id type of 0x0107. When interconnecting with devices from such manufacturers, you must use the **extcommunity-type** command to set the type of router-id to 0x0107.

The type configuration of route-type provides good compatibility with multiple manufacturers. For example, some manufacturers support only the route-type type of 0x8000. When interconnecting with devices from such manufacturers, you must use the **extcommunity-type** command to set the type of route-type to 0x8000.

# Configure the type of router-id of the VRF OSPF protocol to 0x0107.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 10 vrf vrf1
Ruijie(config-router)# extcommunity-type router-id 0107
```

Verifying the BGP/MPLS IP VPN Configurations

This section describes how to verify the L3VPN configurations and VPN routes.

Enter privileged EXEC mode and run the following commands.

Command	Function
Ruijie# **show ip vrf** [*vrf_name*]	Displays the VRF configuration information.
Ruijie# **show bgp vpnv4 unicast all** [*network* | **neighbor** [*peer-address*] | **summary | label]**	Displays the VPN route information.
Ruijie# **show ip route vrf** *vrf_name* [*ip-address* | **bgp | connected | isis | ospf | rip | static]**	Displays the information about the VRF forwarding table.

# Display the VPN route information.

```
Ruijie# show bgp vpnv4 unicast all
Network        Nexthop     Metric  Localprf           Path
Route Distinguisher : 100: 2
*>i 192.168.0.1/32  192.168.0.2  0      100    10 ?
*>i 192.168.1.0/32  192.168.0.2    0      100     ?
Route Distinguisher : 100:30
*>i 192.168.0.1/32  192.168.0.2  0      100    10 ?
*> 192.168.4.0  192.168.4.1         0            20 ?
* 192.168.4.0    0.0.0.0            0      32768     ?
```

Field	Description
*****	Indicates a valid route.
s (lower-case)	Indicates that the route is suppressed by an aggregate route.
s (upper-case)	Indicates that the route is a stale entry.
>	Indicates that the route is an optimal one.
I	Indicates that the route is learned from the IBGP.
Nexthop	Indicates the next hop information of the route.
Metric	Indicates the route metric.
Localprf	Indicates the local preference of the route.
Path	Indicates the AS path included in the route.
I	Indicates that the origin of the route is IGP.
E	Indicates that the origin of the route is EGP.
?	Indicates that the origin of the route is another attribute except IGP or EGP (for example, a redistributed BGP route).

# Display the information about the VRF routing table.

```
Ruijie# show ip route vrf vrf1
Codes: C - connected, S - static, R - RIP,B - BGP
O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2 , ia - IS-IS inter area
* - candidate default
```

```
B    192.168.0.1/32 , [200/0] via 192.168.0.2, 01:02:33
B    192.168.0.3/32 , [200/0] via 192.168.4.1 , 01:02:33
C    192.168.4.0/24 is directly connected ,eth1
```

# Display the VRF configuration information.

```
Ruijie# show ip vrf vrf1
VRF vrf1; default RD : 100: 2
Interfaces:
Eth0
Export VPN route-target communities:
RT : 100: 30
No import VPN route-target community
No import route-map
```

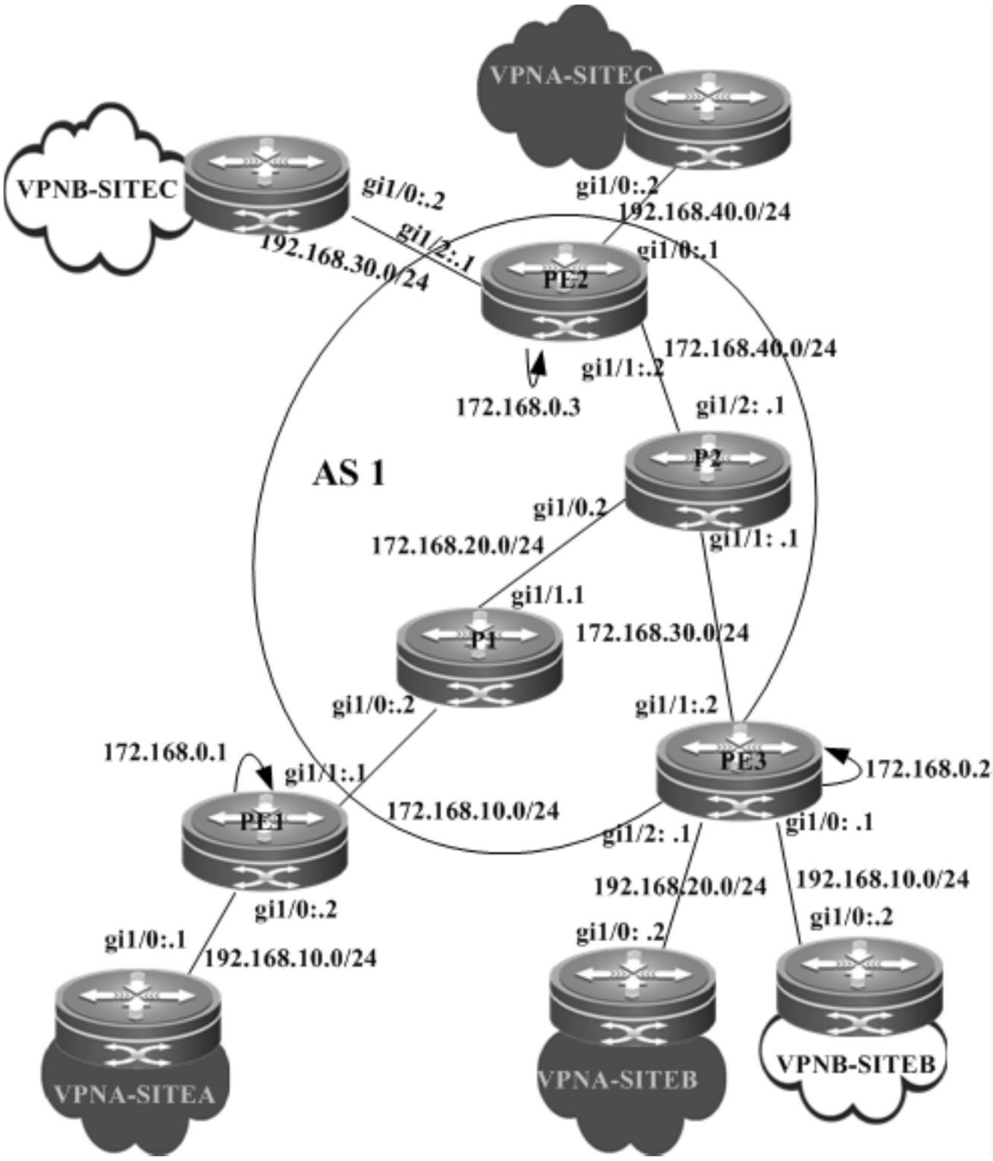
BGP/MPLS IP VPN Configuration Examples

Intranet Configuration Example

Networking Requirements

There are two VPN users: VPNA and VPNB. VPNA has sites at SITEA, SITEB, and SITEC and VPNB has sites at SITEB and SITEC. It is now required that: the users at different sites of VPNA access each other, the users at different sites of VPNB access each other, and the users at the two VPNs not access each other, as shown in the following figure.

Figure 29



Configuration Steps

PE1:

# Configure the loopback interface.

```
Ruijie# configure terminal
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 172.168.0.1 255.255.255.255
```

# Configure the VRF.

Create one VRF instance: VPNA. Set the RD and RT values, and associate the VRF with the corresponding interface.

```
Ruijie# configure terminal
Ruijie(config)# ip vrf VPNA
Ruijie(config-vrf)# rd 1:100
Ruijie(config-vrf)# route-target both 1:100
Ruijie(config-vrf)# end
```

# Associate the VRF with an interface.

```
Ruijie# configure terminal
Ruijie(config)# interface gigabitethernet 1/0
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-GigabitEthernet 1/0)# no switchport
```

# Enable MPLS fast forwarding on a router. This command is unnecessary on a switch.

```
Ruijie(config-if-GigabitEthernet 1/0)# ip ref
Ruijie(config-if-GigabitEthernet 1/0)#ip vrf forwarding VPNA
Ruijie(config-if-GigabitEthernet 1/0)#ip address 192.168.10.2 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/0)# end
```

# Enable BGP and set up MP-IBGP sessions with PE2 and PE3.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 172.168.0.2 remote-as 1
Ruijie(config-router)# neighbor 172.168.0.2 update-source loopback 0
Ruijie(config-router)# neighbor 172.168.0.3 remote-as 1
Ruijie(config-router)# neighbor 172.168.0.3 update-source loopback 0
Ruijie(config-router)# address-family vpnv4
Ruijie(config-router-af)# neighbor 172.168.0.2 activate
Ruijie(config-router-af)# neighbor 172.168.0.3 activate
Ruijie(config-router-af)# end
```

# Configure CE neighbors through EBGP.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config)# address-family ipv4 vrf VPNA
Ruijie(config-router-af)# neighbor 172.168.10.1 remote-as 65002
Ruijie(config-router-af)# end
```

# Configure the MPLS signaling protocol on the backbone network and enable MPLS on the public network interface.

```
Ruijie# configure terminal
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# exit
Ruijie(config)# interface gigabitethernet 1/1
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-GigabitEthernet 1/1)# no switchport
```

# Enable MPLS fast forwarding on a router. This command is unnecessary on a switch.

```
Ruijie(config-if-GigabitEthernet 1/1)# ip ref
Ruijie(config-if-GigabitEthernet 1/1)# ip address 172.168.10.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/1)# label-switching
Ruijie(config-if-GigabitEthernet 1/1)# mpls ip
Ruijie(config-if-GigabitEthernet 1/1)# end
```

# Configure a routing protocol on the backbone network.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 172.168.10.0 0.0.0.255 area 0
Ruijie(config-router)# network 172.168.0.1 0.0.0.0 area 0
Ruijie(config-router)# end
```

PE2:

# Configure the loopback interface.

```
Ruijie# configure terminal
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 172.168.0.3 255.255.255.255
```

# Configure the VRF.

Create two VRFs: VPNA and VPNB. Set the RD and RT values, and associate the VRFs with corresponding interfaces.

```
Ruijie# configure terminal
Ruijie(config)# ip vrf VPNA
Ruijie(config-vrf)# rd 1:100
Ruijie(config-vrf)# route-target both 1:100
Ruijie(config-vrf)# exit
Ruijie(config)# ip vrf VPNB
Ruijie(config-vrf)# rd 1:200
Ruijie(config-vrf)# route-target both 1:200
Ruijie(config-vrf)# exit
```

# Associate the VRF with an interface.

```
Ruijie# configure terminal
Ruijie(config)# interface gigabitethernet 1/0
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-GigabitEthernet 1/0)# no switchport
```

# Enable MPLS fast forwarding on a router. This command is unnecessary on a switch.

```
Ruijie(config-if-GigabitEthernet 1/0)# ip ref
Ruijie(config-if-GigabitEthernet 1/0)# ip vrf forwarding VPNB
Ruijie(config-if-GigabitEthernet 1/0)# ip address 192.168.10.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/0)# exit
Ruijie(config)# interface gigabitethernet 1/1
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-GigabitEthernet 1/1)# no switchport
```

# Enable MPLS fast forwarding on a router. This command is unnecessary on a switch.

```
Ruijie(config-if-GigabitEthernet 1/1)# ip ref
Ruijie(config-if-GigabitEthernet 1/1)#ip vrf forwarding VPNA
Ruijie(config-if-GigabitEthernet 1/1)#ip address 192.168.20.1
255.255.255.0
Ruijie(config-if-GigabitEthernet 1/1)# exit
```

# Enable BGP and set up MP-IBGP sessions with PE2 and PE3.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 172.168.0.1 remote-as 1
Ruijie(config-router)# neighbor 172.168.0.1 update-source loopback 0
Ruijie(config-router)# neighbor 172.168.0.3 remote-as 1
Ruijie(config-router)# neighbor 172.168.0.3 update-source loopback 0
Ruijie(config-router)# address-family vpnv4
Ruijie(config-router-af)# neighbor 172.168.0.1 activate
Ruijie(config-router-af)# neighbor 172.168.0.3 activate
Ruijie(config-router-af)# end
```

# Configure CE neighbors through EBGP.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config)# address-family ipv4 vrf VPNA
Ruijie(config-router-af)# neighbor 172.168.20.2 remote-as 65003
Ruijie(config-router-af)# exit
Ruijie(config)# address-family ipv4 vrf VPNB
Ruijie(config-router-af)# neighbor 172.168.10.2 remote-as 65004
Ruijie(config-router-af)# end
```

# Configure the MPLS signaling protocol on the backbone network and enable MPLS on the public network interface.

```
Ruijie# configure terminal
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# exit
Ruijie(config)# interface gigabitethernet 1/1
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-GigabitEthernet 1/1)# no switchport
```

# Enable MPLS fast forwarding on a router. This command is unnecessary on a switch.

```
Ruijie(config-if-GigabitEthernet 1/1)# ip ref
Ruijie(config-if-GigabitEthernet 1/1)# ip address 172.168.30.2 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/1)# label-switching
Ruijie(config-if-GigabitEthernet 1/1)# mpls ip
Ruijie(config-if-GigabitEthernet 1/1)# end
```

# Configure a routing protocol on the backbone network.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 172.168.30.0 255.255.255.0 area 0
Ruijie(config-router)# network 172.168.0.2 255.255.255.255 area 0
Ruijie(config-router)# end
```

PE3:

The configuration procedure is similar to that of PE2.

VPNA-SITEA:

Set up an EBGP session with PE1.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 65002
Ruijie(config-router)# neighbor 172.168.10.2 remote-as 1
Ruijie(config-router-af)# end
```

VPNA-SITEB:

The configuration procedure is similar to that of VPNA-SITEA.

VPNA-SITEC:

The configuration procedure is similar to that of VPNA-SITEA.

VPNB-SITEB:

The configuration procedure is similar to that of VPNA-SITEA.

VPNB-SITEC:

The configuration procedure is similar to that of VPNA-SITEA.

P1:

# Configure the loopback interface.

```
Ruijie# configure terminal
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 172.168.0.5 255.255.255.255
```

# Configure the MPLS signaling protocol on the backbone network and enable MPLS on the public network interface.

```
Ruijie# configure terminal
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# exit
Ruijie(config)# interface gigabitethernet 1/0
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-GigabitEthernet 1/0)# no switchport
```

# Enable MPLS fast forwarding on a router. This command is unnecessary on a switch.

```
Ruijie(config-if-GigabitEthernet 1/0)# ip ref
Ruijie(config-if-GigabitEthernet 1/0)# ip address 172.168.10.2 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/0)# label-switching
Ruijie(config-if-GigabitEthernet 1/0)# mpls ip
Ruijie(config-if-GigabitEthernet 1/0)# exit
Ruijie(config)# interface gigabitethernet 1/1
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-GigabitEthernet 1/1)# no switchport
```

# Enable MPLS fast forwarding on a router. This command is unnecessary on a switch.

```
Ruijie(config-if-GigabitEthernet 1/1)# ip ref
Ruijie(config-if-GigabitEthernet 1/1)# ip address 172.168.20.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/1)# label-switching
Ruijie(config-if-GigabitEthernet 1/1)# mpls ip
Ruijie(config-if-GigabitEthernet 1/0)# end
```

# Configure a routing protocol on the backbone network.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 172.168.10.0 0.0.0.255 area 0
Ruijie(config-router)# network 172.168.20.0 0.0.0.255 area 0
Ruijie(config-router)# end
```

P2:

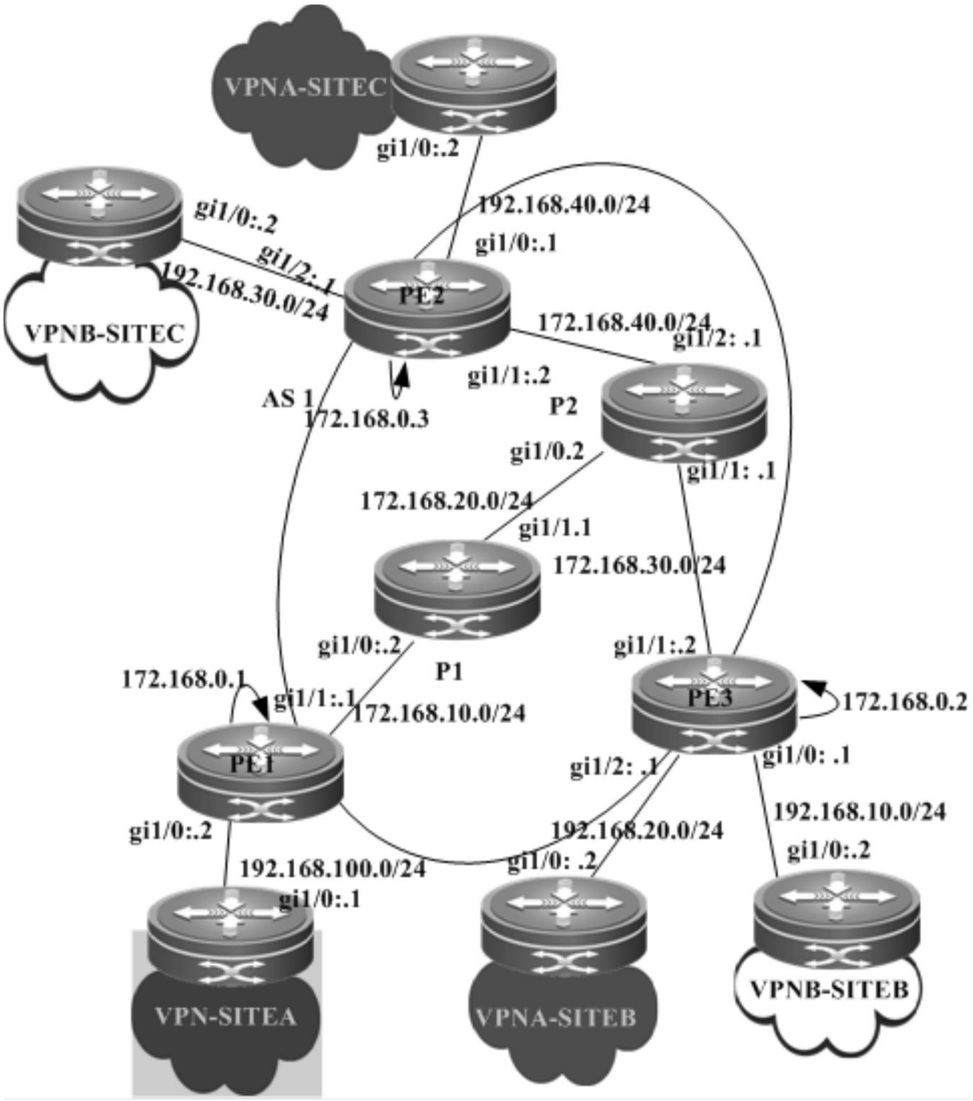
The configuration procedure is similar to that of P1.

Extranet Configuration Example

Networking Requirements

There are two VPN users: VPNA and VPNB. Mutual access is required in a VPN. The two VPNs cannot access each other but can access some shared resources. As shown in the following figure, VPNA and VPNB sites should access the resources of VPN-SITEA.

Figure 30



Configuration Steps

PE1:

# Configure the loopback interface.

```
Ruijie# configure terminal
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 172.168.0.1 255.255.255.255
```

# Configure the VRF.

Create one VRF instance: VPN_EXTRA. Set the RD and RT values, and associate the VRF with the corresponding interface.

```
Ruijie# configure terminal
Ruijie(config)# ip vrf VPN_EXTRA
Ruijie(config-vrf)# rd 1:100
Ruijie(config-vrf)# route-target both 1:100
Ruijie(config-vrf)# route-target both 1:200
Ruijie(config-vrf)# end
```

# Associate the VRF with an interface.

```
Ruijie# configure terminal
Ruijie(config)# interface gigabitethernet 1/0
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-GigabitEthernet 1/0)# no switchport
```

# Enable MPLS fast forwarding on a router. This command is unnecessary on a switch.

```
Ruijie(config-if-GigabitEthernet 1/0)# ip ref
Ruijie(config-if-GigabitEthernet 1/0)# ip vrf forwarding VPN_EXTRA
Ruijie(config-if-GigabitEthernet 1/0)# ip address 192.168.100.2 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/0)# end
```

# Enable BGP and set up MP-IBGP sessions with PE2 and PE3.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 172.168.0.2 remote-as 1
Ruijie(config-router)# neighbor 172.168.0.2 update-source loopback 0
Ruijie(config-router)# neighbor 172.168.0.3 remote-as 1
Ruijie(config-router)# neighbor 172.168.0.3 update-source loopback 0
Ruijie(config-router)# address-family vpnv4
Ruijie(config-router-af)# neighbor 172.168.0.2 activate
Ruijie(config-router-af)# neighbor 172.168.0.3 activate
Ruijie(config-router-af)# end
```

# Enable OSPF to exchange routes with a CE.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 10 vrf VPN_EXTRA
Ruijie(config-router)# network 192.168.100.0 255.255.255.0 area 0
Ruijie(config-router)# redistribute bgp subnets
Ruijie(config-router)# exit
Ruijie(config)# router bgp 1
Ruijie(config-router)# address-family ipv4 vrf VPN_EXTRA
Ruijie(config-router-af)# redistribute ospf 10
Ruijie(config-router-af)# end
```

# Configure the MPLS signaling protocol on the backbone network and enable MPLS on the public network interface.

```
Ruijie# configure terminal
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# exit
Ruijie(config)# interface gigabitethernet 1/1
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-GigabitEthernet 1/1)# no switchport
```

# Enable MPLS fast forwarding on a router. This command is unnecessary on a switch.

```
Ruijie(config-if-GigabitEthernet 1/1)# ip ref
Ruijie(config-if-GigabitEthernet 1/1)# ip address 172.168.10.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/1)# label-switching
Ruijie(config-if-GigabitEthernet 1/1)# mpls ip
Ruijie(config-if-GigabitEthernet 1/1)# end
```

# Configure a routing protocol on the backbone network.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 1
Ruijie(config-router)# network 172.168.10.0 0.0.0.255 area 0
Ruijie(config-router)# network 172.168.0.1 0.0.0.0 area 0
Ruijie(config-router)# end
```

PE2:

# Configure the loopback interface.

```
Ruijie# configure terminal
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 172.168.0.3 255.255.255.255
```

# Configure the VRF.

Create two VRF instances: VPNA and VPNB. Set the RD and RT values, and associate the VRFs with corresponding interfaces.

```
Ruijie# configure terminal
Ruijie(config)# ip vrf VPNA
Ruijie(config-vrf)# rd 1:100
Ruijie(config-vrf)# route-target both 1:100
Ruijie(config-vrf)# exit
Ruijie(config)# ip vrf VPNB
Ruijie(config-vrf)# rd 1:200
Ruijie(config-vrf)# route-target both 1:200
Ruijie(config-vrf)# exit
```

# Associate the VRF with an interface.

```
Ruijie# configure terminal
Ruijie(config)# interface gigabitethernet 1/0
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-GigabitEthernet 1/0)# no switchport
```

# Enable MPLS fast forwarding on a router. This command is unnecessary on a switch.

```
Ruijie(config-if-GigabitEthernet 1/0)# ip ref
Ruijie(config-if-GigabitEthernet 1/0)# ip vrf forwarding VPNB
Ruijie(config-if-GigabitEthernet 1/0)# ip address 192.168.10.1
255.255.255.0
Ruijie(config-if-GigabitEthernet 1/0)# exit
Ruijie(config)# interface gigabitethernet 1/1
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-GigabitEthernet 1/1)# no switchport
```

# Enable MPLS fast forwarding on a router. This command is unnecessary on a switch.

```
Ruijie(config-if-GigabitEthernet 1/1)# ip ref
Ruijie(config-if-GigabitEthernet 1/1)# ip vrf forwarding VPNA
Ruijie(config-if-GigabitEthernet 1/1)# ip address 192.168.20.1
255.255.255.0
Ruijie(config-if-GigabitEthernet 1/1)# exit
```

# Enable BGP and set up MP-IBGP sessions with PE2 and PE3.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 172.168.0.1 remote-as 1
Ruijie(config-router)# neighbor 172.168.0.1 update-source loopback 0
Ruijie(config-router)# neighbor 172.168.0.3 remote-as 1
Ruijie(config-router)# neighbor 172.168.0.3 update-source loopback 0
Ruijie(config-router)# address-family vpnv4
Ruijie(config-router-af)# neighbor 172.168.0.1 activate
Ruijie(config-router-af)# neighbor 172.168.0.3 activate
Ruijie(config-router-af)# end
```

# Enable OSPF to exchange VPN routes with a CE.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 10 vrf VPNA
Ruijie(config-router)# network 192.168.20.0 255.255.255.0 area 0
Ruijie(config-router)# redistribute bgp subnets
Ruijie(config-router)# exit
Ruijie(config)# router ospf 20 vrf VPNB
Ruijie(config-router)# network 192.168.10.0 255.255.255.0 area 0
Ruijie(config-router)# redistribute bgp subnets
Ruijie(config-router)# exit
Ruijie(config)# router bgp 1
Ruijie(config-router)# address-family ipv4 vrf VPNA
Ruijie(config-router-af)# redistribute ospf 10
Ruijie(config-router-af)# exit
Ruijie(config-router)# address-family ipv4 vrf VPNB
Ruijie(config-router-af)# redistribute ospf 20
Ruijie(config-router-af)# exit
```

# Configure the MPLS signaling protocol on the backbone network and enable MPLS on the public network interface.

```
Ruijie# configure terminal
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# exit
Ruijie(config)# interface gigabitethernet 1/1
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-GigabitEthernet 1/1)# no switchport
```

# Enable MPLS fast forwarding on a router. This command is unnecessary on a switch.

```
Ruijie(config-if-GigabitEthernet 1/1)# ip ref
Ruijie(config-if-GigabitEthernet 1/1)# ip address 172.168.30.2 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/1)# label-switching
Ruijie(config-if-GigabitEthernet 1/1)# mpls ip
Ruijie(config-if-GigabitEthernet 1/1)# end
```

# Configure a routing protocol on the backbone network.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 1
Ruijie(config-router)# network 172.168.30.0 0.0.0.255 area 0
Ruijie(config-router)# network 172.168.0.2 0.0.0.0 area 0
Ruijie(config-router)# end
```

PE3:

The configuration procedure is similar to that of PE2.

VPNA-SITEA:

```
Ruijie# configure terminal
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 192.168.100.0 255.255.255.0 area 0
Ruijie(config-router)# end
```

VPNA-SITEB:

The configuration procedure is similar to that of VPNA-SITEA.

VPNA-SITEC:

The configuration procedure is similar to that of VPNA-SITEA.

VPNB-SITEB:

The configuration procedure is similar to that of VPNA-SITEA.

VPNB-SITEC:

The configuration procedure is similar to that of VPNA-SITEA.

P1:

# Configure the loopback interface.

```
Ruijie# configure terminal
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 172.168.0.5 255.255.255.255
```

# Configure the MPLS signaling protocol on the backbone network and enable MPLS on the public network interface.

```
Ruijie# configure terminal
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# exit
Ruijie(config)# interface gigabitethernet 1/0
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-GigabitEthernet 1/0)# no switchport
```

# Enable MPLS fast forwarding on a router. This command is unnecessary on a switch.

```
Ruijie(config-if-GigabitEthernet 1/0)# ip ref
Ruijie(config-if-GigabitEthernet 1/0)# ip address 172.168.10.2 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/0)# label-switching
Ruijie(config-if-GigabitEthernet 1/0)# mpls ip
Ruijie(config-if-GigabitEthernet 1/0)# exit
Ruijie(config)# interface gigabitethernet 1/1
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-GigabitEthernet 1/1)# no switchport
```

# Enable MPLS fast forwarding on a router. This command is unnecessary on a switch.

```
Ruijie(config-if-GigabitEthernet 1/1)# ip ref
Ruijie(config-if-GigabitEthernet 1/1)# ip address 172.168.20.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/1)# label-switching
Ruijie(config-if-GigabitEthernet 1/1)# mpls ip
Ruijie(config-if-GigabitEthernet 1/0)# end
```

# Configure a routing protocol on the backbone network.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 1
Ruijie(config-router)# network 172.168.10.0 0.0.0.255 area 0
Ruijie(config-router)# network 172.168.20.0 0.0.0.255 area 0
Ruijie(config-router)# end
```

P2:

The configuration procedure is similar to that of P1.

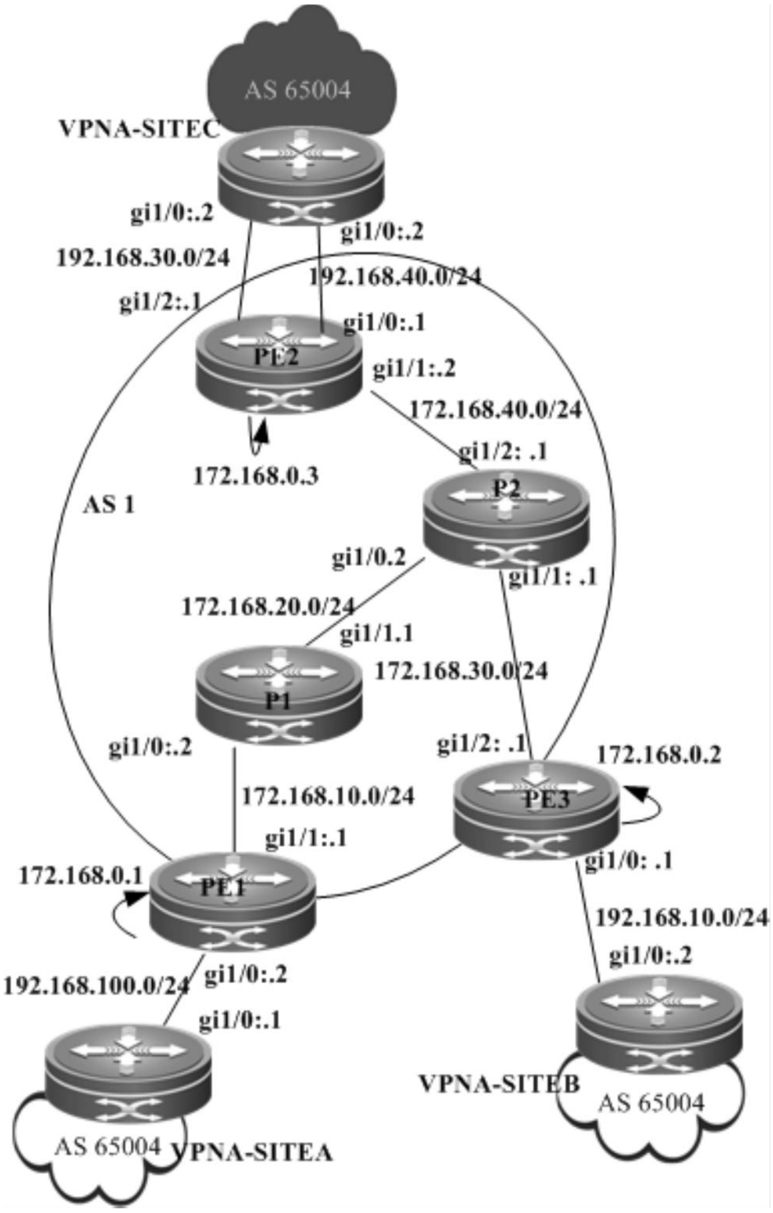
For the protocol between PEs and CEs, you can choose EBGP, OSPF, RIP, or other routing protocol as required.

Hub-and-Spoke Configuration Example

Networking Requirements

The VPN internal data should not be directly exchanged. Instead, the data must be exchanged through the unified control center. Only the control center is entitled to access all resources of a VPN. Any VPN users who want to obtain the VPN resources, must be notified through the control center. As shown in the following figure, to access VPNA-SITEB resources, VPNA-SITEA must pass the control center VPNA-SITEC. Direct access is not available.

Figure 31



Configuration Steps

PE1:

# Configure the loopback interface.

```
Ruijie# configure terminal
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 172.168.0.1 255.255.255.255
```

# Configure the VRF.

Create one VRF instance: spoke1. Set the RD and RT values, and associate the VPNA with the corresponding interface.

```
Ruijie# configure terminal
Ruijie(config)# ip vrf spoke1
Ruijie(config-vrf)# rd 1:100
Ruijie(config-vrf)# route-target export 1:200
Ruijie(config-vrf)# route-target import 1:100
Ruijie(config-vrf)# end
```

# Associate the VRF with an interface.

```
Ruijie# configure terminal
Ruijie(config)# interface gigabitethernet 1/0
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-GigabitEthernet 1/0)# no switchport
```

# Enable MPLS fast forwarding on a router. This command is unnecessary on a switch.

```
Ruijie(config-if-GigabitEthernet 1/0)# ip ref
Ruijie(config-if-GigabitEthernet 1/0)# ip vrf forwarding spoke1
Ruijie(config-if-GigabitEthernet 1/0)# ip address 192.168.100.2 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/0)# end
```

# Enable BGP and set up MP-IBGP sessions with PE3.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 172.168.0.3 remote-as 1
Ruijie(config-router)# neighbor 172.168.0.3 update-source loopback 0
Ruijie(config-router)# address-family vpnv4
Ruijie(config-router-af)# neighbor 172.168.0.3 activate
Ruijie(config-router-af)# neighbor 172.168.0.3 allowas-in
Ruijie(config-router-af)# end
```

# Configure CE neighbors through EBGP.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config)# address-family ipv4 vrf spoke1
Ruijie(config-router-af)# neighbor 192.168.100.1 remote-as 65004
Ruijie(config-router-af)# neighbor 192.168.100.1 as-override
Ruijie(config-router-af)# end
```

# Configure the MPLS signaling protocol on the backbone network and enable MPLS on the public network interface.

```
Ruijie# configure terminal
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# exit
Ruijie(config)# interface gigabitethernet 1/1
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-GigabitEthernet 1/1)# no switchport
```

# Enable MPLS fast forwarding on a router. This command is unnecessary on a switch.

```
Ruijie(config-if-GigabitEthernet 1/1)# no switchport
Ruijie(config-if-GigabitEthernet 1/1)# ip address 172.168.10.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/1)# label-switching
Ruijie(config-if-GigabitEthernet 1/1)# mpls ip
Ruijie(config-if-GigabitEthernet 1/1)# end
```

# Configure a routing protocol on the backbone network.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 172.168.10.0 0.0.0.255 area 0
Ruijie(config-router)# network 172.168.0.1 0.0.0.0 area 0
Ruijie(config-router)# end
```

PE2:

# Configure the loopback interface.

```
Ruijie# configure terminal
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 172.168.0.3 255.255.255.255
```

# Configure the VRF.

Create one VRF instance: spoke2. Set the RD and RT values, and associate the VRF with the corresponding interface.

```
Ruijie# configure terminal
Ruijie(config)# ip vrf spoke2
Ruijie(config-vrf)# rd 1:100
Ruijie(config-vrf)# route-target export 1:300
Ruijie(config-vrf)# route-target import 1:100
Ruijie(config-vrf)# exit
```

# Associate the VRF with an interface.

```
Ruijie# configure terminal
Ruijie(config)# interface gigabitethernet 1/0
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-GigabitEthernet 1/0)# no switchport
```

# Enable MPLS fast forwarding on a router. This command is unnecessary on a switch.

```
Ruijie(config-if-GigabitEthernet 1/0)# no switchport
Ruijie(config-if-GigabitEthernet 1/0)# ip vrf forwarding spoke2
Ruijie(config-if-GigabitEthernet 1/0)# ip address 192.168.10.1
255.255.255.0
Ruijie(config-if-GigabitEthernet 1/0)# exit
```

# Enable BGP and set up MP-IBGP sessions with PE3.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 172.168.0.3 remote-as 1
Ruijie(config-router)# neighbor 172.168.0.3 update-source loopback 0
Ruijie(config-router)# address-family vpnv4
Ruijie(config-router-af)# neighbor 172.168.0.3 activate
Ruijie(config-router-af)# neighbor 172.168.0.3 allowas-in
Ruijie(config-router-af)# end
```

# Configure CE neighbors through EBGP.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# address-family ipv4 vrf spoke2
Ruijie(config-router-af)# neighbor 192.168.10.2 remote-as 65004
Ruijie(config-router-af)# neighbor 192.168.10.2 as-override
Ruijie(config-router-af)# end
```

# Configure the MPLS signaling protocol on the backbone network and enable MPLS on the public network interface.

```
Ruijie# configure terminal
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# exit
Ruijie(config)# interface gigabitethernet 1/1
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-GigabitEthernet 1/1)# no switchport
```

# Enable MPLS fast forwarding on a router. This command is unnecessary on a switch.

```
Ruijie(config-if-GigabitEthernet 1/1)# no switchport
Ruijie(config-if-GigabitEthernet 1/1)# ip address 172.168.30.2 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/1)# label-switching
Ruijie(config-if-GigabitEthernet 1/1)# mpls ip
Ruijie(config-if-GigabitEthernet 1/1)# end
```

# Configure a routing protocol on the backbone network.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 172.168.30.0 0.0.0.255 area 0
Ruijie(config-router)# network 172.168.0.2 0.0.0.0 area 0
Ruijie(config-router)# end
```

PE3:

# Configure the loopback interface.

```
Ruijie# configure terminal
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 172.168.0.2 255.255.255.255
```

# Configure the VRF.

Create two VRF instances: from-spoke and from-hub. Set the RD and RT values, and associate the VRFs with corresponding interfaces.

```
Ruijie# configure terminal
Ruijie(config)# ip vrf from-spoke
Ruijie(config-vrf)# rd 1:100
Ruijie(config-vrf)# route-target import 1:300
Ruijie(config-vrf)# route-target import 1:200
Ruijie(config-vrf)# exit
Ruijie(config)# ip vrf from-hub
Ruijie(config-vrf)# rd 1:200
Ruijie(config-vrf)# route-target export 1:100
Ruijie(config-vrf)# exit
```

# Associate the VRF with an interface.

```
Ruijie# configure terminal
Ruijie(config)# interface gigabitethernet 1/0
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-GigabitEthernet 1/0)# no switchport
```

# Enable MPLS fast forwarding on a router. This command is unnecessary on a switch.

```
Ruijie(config-if-GigabitEthernet 1/0)# no switchport
Ruijie(config-if-GigabitEthernet 1/0)# ip vrf forwarding from-hub
Ruijie(config-if-GigabitEthernet 1/0)# ip address 192.168.40.1
255.255.255.0
Ruijie(config-if-GigabitEthernet 1/0)# exit
Ruijie(config)# interface gigabitethernet 1/1
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-GigabitEthernet 1/1)# no switchport
```

# Enable MPLS fast forwarding on a router. This command is unnecessary on a switch.

```
Ruijie(config-if-GigabitEthernet 1/1)# no switchport
Ruijie(config-if-GigabitEthernet 1/1)# ip vrf forwarding from-spoke
Ruijie(config-if-GigabitEthernet 1/1)# ip address 192.168.30.1
255.255.255.0
Ruijie(config-if-GigabitEthernet 1/1)# exit
```

# Enable BGP and set up MP-IBGP sessions with PE1 and PE2.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 172.168.0.1 remote-as 1
Ruijie(config-router)# neighbor 172.168.0.1 update-source loopback 0
Ruijie(config-router)# neighbor 172.168.0.2 remote-as 1
Ruijie(config-router)# neighbor 172.168.0.2 update-source loopback 0
Ruijie(config-router)# address-family vpnv4
Ruijie(config-router-af)# neighbor 172.168.0.3 activate
Ruijie(config-router-af)# neighbor 172.168.0.2 activate
Ruijie(config-router-af)# end
```

# Configure CE neighbors through EBGP.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# address-family ipv4 vrf from-spoke
Ruijie(config-router-af)# neighbor 192.168.30.2 remote-as 65004
Ruijie(config-router-af)# neighbor 192.168.30.2 as-override
Ruijie(config-router-af)# exit
Ruijie(config-router)# address-family ipv4 vrf from-hub
Ruijie(config-router-af)# neighbor 192.168.40.2 remote-as 65004
Ruijie(config-router-af)# neighbor 192.168.40.2 allows-in
Ruijie(config-router-af)# exit
```

# Configure the MPLS signaling protocol on the backbone network and enable MPLS on the public network interface.

```
Ruijie# configure terminal
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# exit
Ruijie(config)# interface gigabitethernet 1/1
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-GigabitEthernet 1/1)# no switchport
```

# Enable MPLS fast forwarding on a router. This command is unnecessary on a switch.

```
Ruijie(config-if-GigabitEthernet 1/1)# no switchport
Ruijie(config-if-GigabitEthernet 1/1)# ip address 172.168.30.2 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/1)# label-switching
Ruijie(config-if-GigabitEthernet 1/1)# mpls ip
Ruijie(config-if-GigabitEthernet 1/1)# end
```

# Configure a routing protocol on the backbone network.

```
Ruijie# configure terminal
```

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 172.168.30.0 0.0.0.255 area 0
Ruijie(config-router)# network 172.168.0.2 0.0.0.0 area 0
Ruijie(config-router)# end
```

VPNA-SITEA:

# Configure a PE session through EBGP.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 65004
Ruijie(config-router)# neighbor 192.168.100.2 remote-as 1
Ruijie(config-router)# exit
```

The configuration of VPNA-SITEB is similar to that of VPNA-SITEA.

VPNA-SITEC:

# Configure a PE session through EBGP.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 65004
Ruijie(config-router)# neighbor 192.168.30.1 remote-as 1
Ruijie(config-router)# neighbor 192.168.40.1 remote-as 1
Ruijie(config-router)# exit
```

Internet Unified Egress Interface Configuration Example

Networking Requirements

Several VPNs cannot access each other, but the VPNs need to access the Internet through a device. As shown in Figure 32 and Figure 33, VPN1 and VPN2 cannot access each other but can access the Internet through PE1.

To isolate VPN1 and VPN2, you can use centralized isolation or distributed isolation. The differences between the above schemes are as follows:

For the centralized isolation, add filtering rules on only a CE egress interface for subsequently added VPN sites to access the Internet through the unified egress interface. Other VPN sites do not need any change of configuration. This scheme features good scalability. The disadvantage of this scheme is that isolated packets can be discarded only when they reach the CE egress interface, which consumes network bandwidth.

For the distributed isolation, add filtering rules on the CE of each VPN site that needs to access the Internet through the unified egress interface. This scheme features poor scalability. The advantage of this scheme is that isolated packets are discarded on the CE of a VPN site, which saves network bandwidth.

Networking Requirements

Several VPNs cannot access each other, but the VPNs need to access the Internet through a device. As shown in Figure 32 and Figure 33, VPN1 and VPN2 cannot access each other but can access the Internet through PE1.

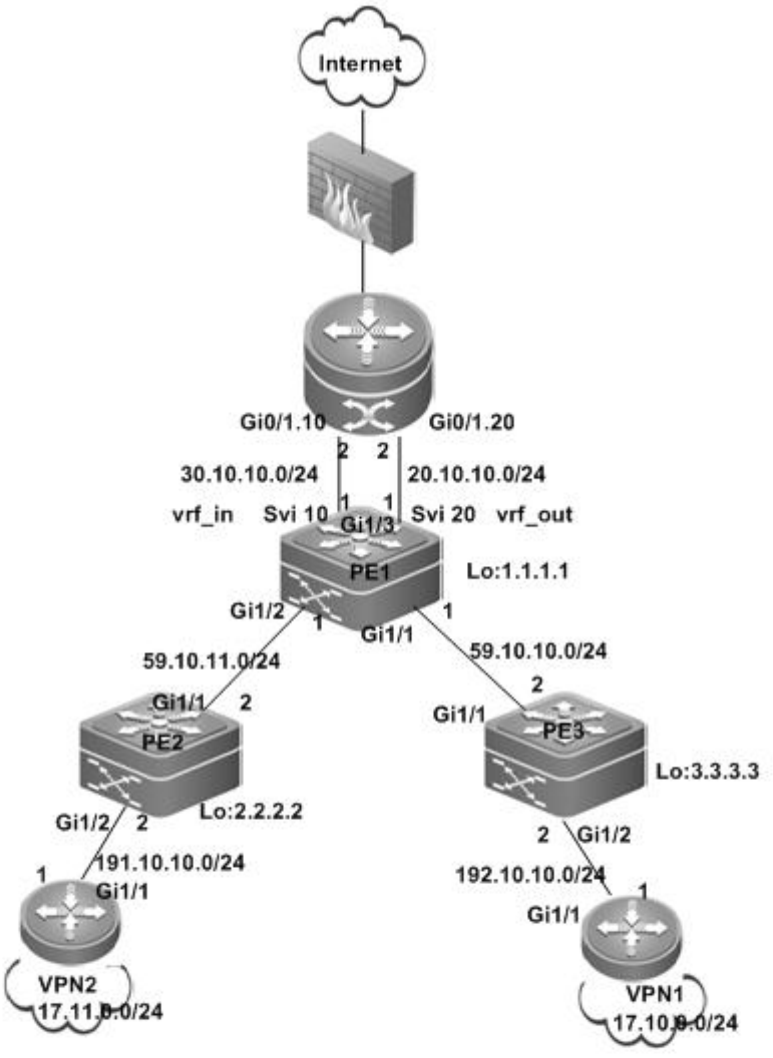
To isolate VPN1 and VPN2, you can use centralized isolation or distributed isolation. The differences between the above schemes are as follows:

For the centralized isolation, add filtering rules on only a CE egress interface for subsequently added VPN sites to access the Internet through the unified egress interface. Other VPN sites do not need any change of configuration. This scheme features good scalability. The disadvantage of this scheme is that isolated packets can be discarded only when they reach the CE egress interface, which consumes network bandwidth.

For the distributed isolation, add filtering rules on the CE of each VPN site that needs to access the Internet through the unified egress interface. This scheme features poor scalability. The advantage of this scheme is that isolated packets are discarded on the CE of a VPN site, which saves network bandwidth.

Centralized Isolation

Figure 32



Configuration Steps

- Configuring PE2

# Configure the loopback interface.

```
Ruijie# configure terminal
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 2.2.2.2 255.255.255.255
```

# Configure the VRF.

Create a VRF, create VPN1, and set the RD and RT values.

```
Ruijie# configure terminal
Ruijie(config)# ip vrf VPN1
Ruijie(config-vrf)# rd 1:100
Ruijie(config-vrf)# route-target both 1:100
Ruijie(config-vrf)# end
```

# Associate the VRF with an interface.

```
Ruijie# configure terminal
Ruijie(config)# interface gigabitethernet 1/2
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-GigabitEthernet 1/2)# no switchport
Ruijie(config-if-GigabitEthernet 1/2)# ip vrf forwarding VPN1
Ruijie(config-if-GigabitEthernet 1/2)# ip address 191.10.10.2
255.255.255.0
Ruijie(config-if-GigabitEthernet 1/2)# end
```

# Configure OSPF route exchange with CE2.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 10 vrf VPN2
Ruijie(config-router)# network 191.10.10.0 0.0.0.255 area 0
Ruijie(config-router)# default-information originate
Ruijie(config-router)# redistribute bgp subnets
Ruijie(config-router)# exit
```

# Establish an IBGP neighbor relation with PE1.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 1.1.1.1 remote-as 1
Ruijie(config-router)# neighbor 1.1.1.1 update-source loopback 0
Ruijie(config-router)# address-family vpnv4
Ruijie(config-router-af)# neighbor 1.1.1.1 activate
Ruijie(config-router-af)# end
```

# Configure IP route distribution.

```
Ruijie(config)# router bgp 1
Ruijie(config-router)# address-family ipv4 vrf VPN1
Ruijie(config-router-af)# redistribute ospf 10
Ruijie(config-router-af)# end
```

# Configure MPLS signaling for the backbone network and enable MPLS on the public network interface.

```
Ruijie# configure terminal
```

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# exit
Ruijie(config)# interface gigabitethernet 1/1
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-GigabitEthernet 1/1)# no switchport
Ruijie(config-if-GigabitEthernet 1/1)# ip address 59.10.11.2
255.255.255.0
Ruijie(config-if-GigabitEthernet 1/1)# label-switching
Ruijie(config-if-GigabitEthernet 1/1)# mpls ip
Ruijie(config-if-GigabitEthernet 1/1)# end
```

# Configure a routing protocol on the backbone network.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 1
Ruijie(config-router)# network 59.10.11.0 0.0.0.255 area 0
Ruijie(config-router)# network 2.2.2.2 0.0.0.0 area 0
Ruijie(config-router)# end
```

- Configuring CE2

# Configure the IP address of an interface connected to PE2.

```
Ruijie# configure terminal
Ruijie(config)# interface gigabitethernet 1/1
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-GigabitEthernet 1/1)# no switchport
Ruijie(config-if-GigabitEthernet 1/1)# ip address 191.10.10.1 255.255.255.0
```

# Configure OSPF route exchange with PE2.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 1
Ruijie(config-router)# network 191.10.10.0 0.0.0.255 area 0
Ruijie(config-router)# network 172.11.0.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

- Configuring PE3

The configuration is similar to the PE2 configuration. Assume that the export and import RT of VPN2 is 1: 200.

- Configuring PE1

# Configure the loopback interface.

```
Ruijie# configure terminal
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 1.1.1.1 255.255.255.255
```

# Create the trunk interface.

```
Ruijie# configure terminal
Ruijie(config)# interface gigabitethernet 1/3
Ruijie(config)# switch mode trunk
Ruijie(config)# switch trunk vlan allow remove vlan 1-9,11-19,21-4094
```

# Configure the out_vrf as the outgoing interface for accessing the Internet.

Create a VRF, configure the vrf_out, set the RD and RT values, and associate the VRF with the corresponding interface.

```
Ruijie# configure terminal
Ruijie(config)# ip vrf vrf_out
Ruijie(config-vrf)# rd 1:300
Ruijie(config-vrf)# route-target export 1:100
Ruijie(config-vrf)# route-target export 1:200
Ruijie(config-vrf)# end
```

# Bind the interface to the VRF.

```
Ruijie# configure terminal
Ruijie(config)#interface vlan 20
Ruijie(config-if-Vlan 20)# ip vrf forwarding vrf_out
Ruijie(config-if-Vlan 20)# ip address 20.10.10.1 255.255.255.0
```

# Configure the default route to the Internet.

```
Ruijie# configure terminal
Ruijie(config)# ip route vrf vrf_out 0.0.0.0 0.0.0.0 vlan 20 20.10.10.2
```

# Configure in_vrf as the incoming interface of traffic returned from the Internet.

Create a VRF, configure the vrf_in, set the RD and RT values, and associate the VRF with the corresponding interface.

```
Ruijie# configure terminal
Ruijie(config)# ip vrf vrf_in
Ruijie(config-vrf)# rd 1:400
Ruijie(config-vrf)# route-target import 1:100
Ruijie(config-vrf)# route-target import 1:200
Ruijie(config-vrf)# end
```

# Bind the interface to the VRF.

```
Ruijie# configure terminal
Ruijie(config)#interface vlan 10
Ruijie(config-if-Vlan 20)# ip vrf forwarding vrf_in
Ruijie(config-if-Vlan 20)# ip address 30.10.10.1 255.255.255.0
```

# Configure EBGP route exchange with CE1.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# address-family ipv4 vrf vrf_in
Ruijie(config-router-af)# neighbor 30.10.10.2 remote-as 100
```

# Establish IBGP neighbor relations with PE2 and PE3.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 2.2.2.2 remote-as 1
Ruijie(config-router)# neighbor 2.2.2.2 update-source loopback 0
Ruijie(config-router)# neighbor 3.3.3.3 remote-as 1
Ruijie(config-router)# neighbor 3.3.3.3 update-source loopback 0
Ruijie(config-router)# address-family vpnv4
Ruijie(config-router-af)# neighbor 2.2.2.2 activate
Ruijie(config-router-af)# neighbor 3.3.3.3 activate
Ruijie(config-router-af)# end
```

# Configure VRF vrf_out route exchange.

```
Ruijie(config)# router bgp 1
Ruijie(config-router)# address-family ipv4 vrf vrf_out
Ruijie(config-router-af)# default-information originate
Ruijie(config-router-af)# redistribute static
Ruijie(config-router-af)# end
```

# Configure MPLS signaling for the backbone network and enable MPLS on the public network interface.

```
Ruijie# configure terminal
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# exit
Ruijie(config)# interface gigabitethernet 1/2
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-GigabitEthernet 1/2)# no switchport
Ruijie(config-if-GigabitEthernet 1/2)# ip address 59.10.11.1
255.255.255.0
Ruijie(config-if-GigabitEthernet 1/2)# label-switching
Ruijie(config-if-GigabitEthernet 1/2)# mpls ip
Ruijie(config-if-GigabitEthernet 1/2)# end
Ruijie(config)# interface gigabitethernet 1/1
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-GigabitEthernet 1/1)# no switchport
Ruijie(config-if-GigabitEthernet 1/1)# ip address 59.10.10.1
255.255.255.0
Ruijie(config-if-GigabitEthernet 1/1)# label-switching
Ruijie(config-if-GigabitEthernet 1/1)# mpls ip
Ruijie(config-if-GigabitEthernet 1/1)# end
```

# Configure a routing protocol on the backbone network.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 1
Ruijie(config-router)# network 59.10.11.0 0.0.0.255 area 0
Ruijie(config-router)# network 59.10.10.0 0.0.0.255 area 0
Ruijie(config-router)# network 1.1.1.1 0.0.0.0 area 0
Ruijie(config-router)# end
```

- Configuring CE1

# Create the sub-interface.

```
Ruijie# configure terminal
Ruijie(config)# interface gigabitethernet 0/1.10
Ruijie(config-if-Gigabitethernet 0/1.10)# encapsulation dot1q 10
Ruijie(config-if-Gigabitethernet 0/1.10)# ip address 30.10.10.2 255.255.255.0
Ruijie(config)# exit
Ruijie(config)# interface gigabitethernet 0/1.20
Ruijie(config-if-Gigabitethernet 0/1.20)# encapsulation dot1q 20
Ruijie(config-if-Gigabitethernet 0/1.20)# ip address 20.10.10.2 255.255.255.0
```

# Establish an EBGP neighbor relation with PE1.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 100
Ruijie(config)# neighbor 30.10.10.1 remote-as 1
```

# Create an ACL rule.

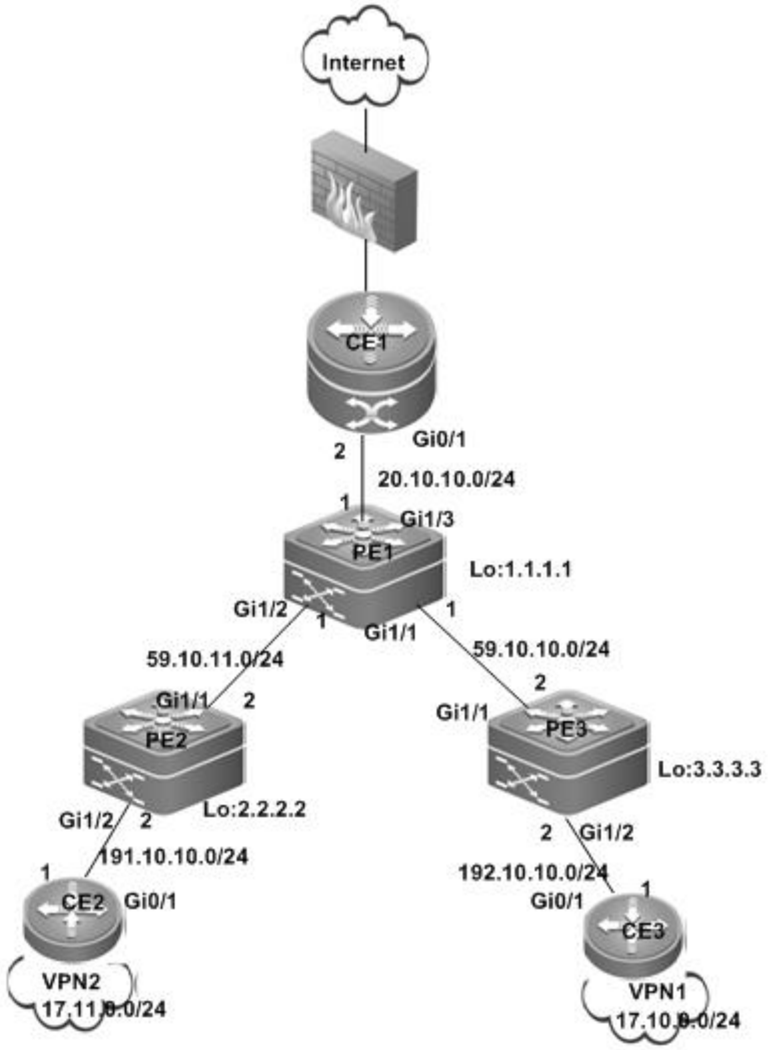
```
Ruijie(config)#access-list 1 deny 17.0.0.0 0.255.255.255
Ruijie(config)# access-list 1 permit any
```

# Configure the ACL rule on the sub-interface Gi 0/1.10.

```
Ruijie(config)# interface gigabitethernet 0/1.10
Ruijie(config-if-Gigabitethernet 0/1.10)# ip access-group 1 out
```

Distributed Isolation

Figure 33



- Configuring PE1

# Configure the loopback interface.

This configuration is the same as Centralized Isolation.

# Configure the VRF.

Create a VRF, create vrf_net, and set the RD and RT values.

```
Ruijie# configure terminal
Ruijie(config)# ip vrf vrf_net
Ruijie(config-vrf)# rd 1:300
Ruijie(config-vrf)# route-target import 1:100
Ruijie(config-vrf)# route-target import 1:200
Ruijie(config-vrf)# route-target export 1:100
Ruijie(config-vrf)# route-target export 1:200
```

# Associate the VRF with an interface.

```
Ruijie# configure terminal
Ruijie(config)# interface gigabitethernet 1/3
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-GigabitEthernet 1/3)# no switchport
Ruijie(config-if-GigabitEthernet 1/3)# ip vrf forwarding vrf_net
Ruijie(config-if-GigabitEthernet 1/3)# ip address 20.10.10.1 255.255.255.0
```

# Configure the default route to the Internet.

```
Ruijie(config)#ip route vrf vrf_net 0.0.0.0 0.0.0.0 gigabitethernet 1/3 20.10.10.2
```

# Configure EBGP route exchange with CE1.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# address-family ipv4 vrf vrf_net
Ruijie(config-router-af)# neighbor 20.10.10.2 remote-as 100
```

# Establish IBGP neighbor relations with PE2 and PE3.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 2.2.2.2 remote-as 1
Ruijie(config-router)# neighbor 2.2.2.2 update-source loopback 0
Ruijie(config-router)# neighbor 3.3.3.3 remote-as 1
Ruijie(config-router)# neighbor 3.3.3.3 update-source loopback 0
Ruijie(config-router)# address-family vpnv4
Ruijie(config-router-af)# neighbor 2.2.2.2 activate
Ruijie(config-router-af)# neighbor 3.3.3.3 activate
Ruijie(config-router-af)# end
```

# Configure VRF vrf_net route exchange.

```
Ruijie(config)# router bgp 1
Ruijie(config-router)# address-family ipv4 vrf vrf_net
Ruijie(config-router-af)# default-information originate
Ruijie(config-router-af)# redistribute static
Ruijie(config-router-af)# end
```

# Configure MPLS and routes for the backbone network.

This configuration is the same as Centralized Isolation **错误!未指定书签。**.

- Configuring CE1

```
Ruijie(config)# interface gigabitethernet 0/1
Ruijie(config-if-Gigabitethernet 0/1)# ip address 20.10.10.2 255.255.255.0
```

# Establish an EBGP neighbor relation with PE1.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 100
Ruijie(config-router)# neighbor 20.10.10.1 remote-as 1
```

- Configuring PE2

This configuration is the same as Centralized Isolation 错误!未指定书签。.

- Configuring CE2

# Configure an ACL rule.

```
Ruijie# configure terminal
Ruijie(config)# access-list 2000 deny ip any 17.10.0.0 0.0.255.255
Ruijie(config)# access-list 2000 permit ip any any
```

# Apply the ACL rule to the interface.

```
Ruijie(config)# interface gigabitethernet 0/1
Ruijie(config-if-Gigabitethernet 0/1)# ip access-group 2000 out
```

# Configure route exchange with PE2.

This configuration is the same as Centralized Isolation.

# Configure the default static route.

This configuration is the same as Centralized Isolation.

Configuring PE3

This configuration is the same as Centralized Isolation.

Configuring CE3

# Configure an ACL rule.

```
Ruijie# configure terminal
Ruijie(config)# access-list 2000 deny ip any 17.11.0.0 0.0.255.255
Ruijie(config)# access-list 2000 permit ip any any
```

# Apply the ACL rule to the interface.

```
Ruijie(config)# interface gigabitethernet 0/1
Ruijie(config-if-Gigabitethernet 0/1)# ip access-group 2000 out
```

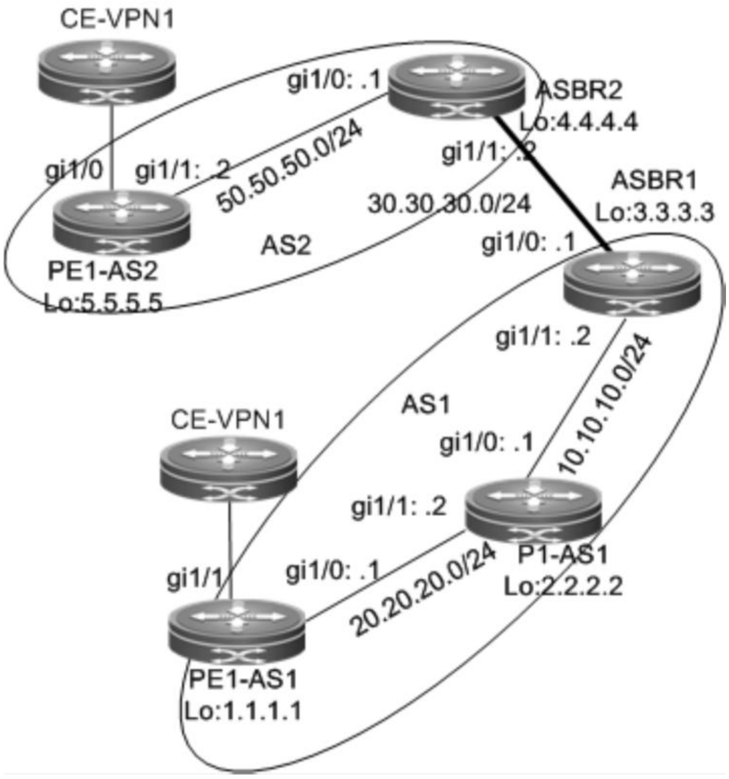
# Other configurations are the same as Centralized Isolation.

Inter-AS VPN OptionB: Next Hop Unchanged

Networking Requirements

One VPN user has sites at both ASs. It is required that the VPN sites in different ASs access each other.

Figure 34 OptionB: Next Hop Unchanged



The configuration scheme is as follows:

PE1-AS1:

# Configure the loopback interface.

```
Ruijie# configure terminal
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 1.1.1.1 255.255.255.255
```

# Configure the VRF.

Create one VRF instance: VPN1. Set the RD and RT values, and associate the VRF with the corresponding interface.

```
Ruijie# configure terminal
Ruijie(config)# ip vrf VPN1
Ruijie(config-vrf)# rd 1:100
Ruijie(config-vrf)# route-target both 1:100
Ruijie(config-vrf)# end
```

# Associate the VRF with an interface.

```
Ruijie# configure terminal
Ruijie(config)# interface gigabitethernet 1/1
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-GigabitEthernet 1/1)# no switchport
Ruijie(config-if-GigabitEthernet 1/1)# ip vrf forwarding VPN1
Ruijie(config-if-GigabitEthernet 1/1)# ip address 192.168.16.2
255.255.255.0
Ruijie(config-if-GigabitEthernet 1/1)# end
```

# Enable BGP and set up MP-IBGP sessions with ASBR1.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 3.3.3.3 remote-as 1
Ruijie(config-router)# neighbor 3.3.3.3 update-source loopback 0
Ruijie(config-router)# address-family vpnv4
Ruijie(config-router-af)# neighbor 3.3.3.3 activate
Ruijie(config-router-af)# end
```

# Configure CE neighbors through EBGP.

See the configuration procedure in the "Running BGP Between PEs and CEs to Transmit Route Information" section and the related configurations in the "Intranet Configuration Examples" section.

# Configure the MPLS signaling protocol on the backbone network and enable MPLS on the public network interface.

```
Ruijie# configure terminal
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# exit
Ruijie(config)# interface gigabitethernet 1/0
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-GigabitEthernet 1/0)# no switchport
Ruijie(config-if-GigabitEthernet 1/0)# ip address 20.20.20.1
255.255.255.0
Ruijie(config-if-GigabitEthernet 1/0)# label-switching
Ruijie(config-if-GigabitEthernet 1/0)# mpls ip
Ruijie(config-if-GigabitEthernet 1/0)# end
```

# Configure a routing protocol on the backbone network.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 20.20.20.0 0.0.0.255 area 0
Ruijie(config-router)# network 1.1.1.1 0.0.0.0 area 0
Ruijie(config-router)# end
```

The configuration procedure of **PE1-AS2** is similar to the preceding one.

P1-AS1:

# Configure the loopback interface.

```
Ruijie# configure terminal
```

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 2.2.2.2 255.255.255.255
```

# Configure the MPLS signaling protocol on the backbone network and enable MPLS on the public network interface.

```
Ruijie# configure terminal
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# exit
Ruijie(config)# interface gigabitethernet 1/0
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-GigabitEthernet 1/0)# no switchport
Ruijie(config-if-GigabitEthernet 1/0)# ip address 10.10.10.1
255.255.255.0
Ruijie(config-if-GigabitEthernet 1/0)# label-switching
Ruijie(config-if-GigabitEthernet 1/0)# mpls ip
Ruijie(config-if-GigabitEthernet 1/0)# exit
Ruijie(config)# interface gigabitethernet 1/1
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-GigabitEthernet 1/1)# no switchport
Ruijie(config-if-GigabitEthernet 1/1)# ip address 20.20.20.2
255.255.255.0
Ruijie(config-if-GigabitEthernet 1/1)# label-switching
Ruijie(config-if-GigabitEthernet 1/1)# mpls ip
```

# Configure a routing protocol on the backbone network.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 20.20.20.0 0.0.0.255 area 0
Ruijie(config-router)# network 10.10.10.0 0.0.0.255 area 0
Ruijie(config-router)# network 2.2.2.2 0.0.0.0 area 0
Ruijie(config-router)# end
```

ASBR1:

# Configure the loopback interface.

```
Ruijie# configure terminal
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 3.3.3.3 255.255.255.255
```

# Enable BGP, disable the BGP RT filtering function, and establish neighbor relations with PE1-AS1 and ASBR2.

```
Ruijie# configure terminal
```

```
Ruijie(config)# router bgp 1
Ruijie(config-router)# no bgp default route-target filter
Ruijie(config-router)# neighbor 1.1.1.1 remote-as 1
Ruijie(config-router)# neighbor 1.1.1.1 update-source loopback 0
Ruijie(config-router)# neighbor 30.30.30.2 remote-as 2
Ruijie(config-router)# address-family vpnv4 unicast
Ruijie(config-router-af)# neighbor 1.1.1.1 activate
Ruijie(config-router-af)# neighbor 30.30.30.2 activate
Ruijie(config-router-af)# end
```

# Configure MPLS signaling and enable MPLS on a public network interface.

```
Ruijie# configure terminal
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# exit
Ruijie(config)# interface gigabitethernet 1/1
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-GigabitEthernet 1/1)# no switchport
Ruijie(config-if-gigabitethernet 1/1)# ip address 10.10.10.2
255.255.255.0
Ruijie(config-if-gigabitethernet 1/1)# label-switching
Ruijie(config-if-gigabitethernet 1/1)# mpls ip
Ruijie(config-if-gigabitethernet 1/1)# end
```

# Run OSPF on the backbone network to transmit routes and redistribute directly connected subnet routes.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 10.10.10.0 0.0.0.255 area 0
Ruijie(config-router)# network 3.3.3.3 0.0.0.0 area 0
Ruijie(config-router)# redisteibute connected subnets
Ruijie(config-router)# end
```

# Assign an IP address to the interface connected to ASBR2.

```
Ruijie(config)# interface gigabitethernet 1/0
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-GigabitEthernet 1/0)# no switchport
Ruijie(config-if-gigabitethernet 1/0)# ip address 30.30.30.1
255.255.255.0
```

# Enable label switching on an interface.

```
Ruijie(config-if-gigabitethernet 1/0)# label-switching
```

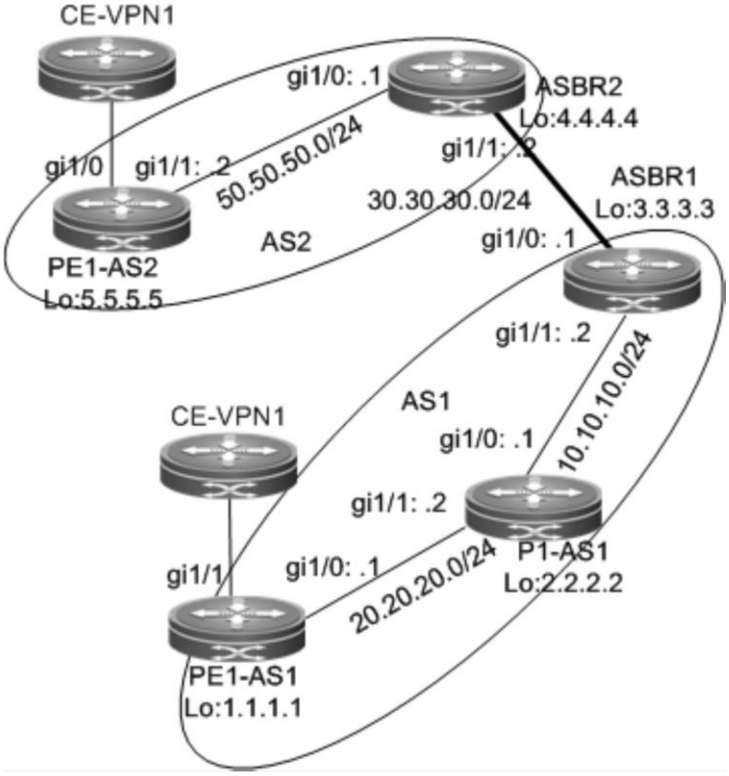
The configuration scheme of ASBR2 is similar to that of ASBR1.

Inter-AS VPN OptionB: Next Hop Changed

Networking Requirements

One VPN user has sites at both ASs. It is required that the VPN sites in different ASs access each other.

Figure 35 OptionB: Next Hop Changed



The configuration scheme is as follows:

PE1-AS1:

The configuration procedure is similar to that of PE1-AS1 in the "Inter-AS VPN OptionB: Next Hop Unchanged" section and is not described here.

P1-AS1:

The configuration procedure is similar to that of P1-AS1 in the "Inter-AS VPN OptionB: Next Hop Unchanged" section and is not described here.

ASBR1:

# Configure the loopback interface.

```
Ruijie# configure terminal
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 3.3.3.3 255.255.255.255
```

# Enable BGP, disable the BGP RT filtering function, establish neighbor relations with the PE and ASBR, and change the next hop of routes to the neighbor PE into the local address.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# no bgp default route-target filter
Ruijie(config-router)# neighbor 1.1.1.1 remote-as 1
Ruijie(config-router)# neighbor 1.1.1.1 update-source loopback 0
Ruijie(config-router)# neighbor 30.30.30.2 remote-as 2
Ruijie(config-router)# address-family vpnv4 unicast
Ruijie(config-router-af)# neighbor 1.1.1.1 activate
Ruijie(config-router-af)# neighbor 1.1.1.1 next-hop-self
Ruijie(config-router-af)# neighbor 30.30.30.2 activate
Ruijie(config-router-af)# end
```

# Configure MPLS signaling and enable MPLS on a public network interface.

```
Ruijie# configure terminal
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# exit
Ruijie(config)# interface gigabitethernet 1/1
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-gigabitethernet 1/1)# no switchport
Ruijie(config-if-gigabitethernet 1/1)# ip address 10.10.10.2
255.255.255.0
Ruijie(config-if-gigabitethernet 1/1)# label-switching
Ruijie(config-if-gigabitethernet 1/1)# mpls ip
Ruijie(config-if-gigabitethernet 1/1)# end
```

# Run OSPF on the backbone network to transmit route information.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 10.10.10.0 0.0.0.255 area 0
Ruijie(config-router)# network 3.3.3.3 0.0.0.0 area 0
Ruijie(config-router)# end
```

# Assign an IP address to the interface connected to ASBR2.

```
Ruijie(config)# interface gigabitethernet 1/0
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-GigabitEthernet 1/0)# no switchport
Ruijie(config-if-gigabitethernet 1/0)# ip address 30.30.30.1
255.255.255.0
```

# Enable label switching on an interface.

```
Ruijie(config-if-gigabitethernet 1/0)# label-switching
```

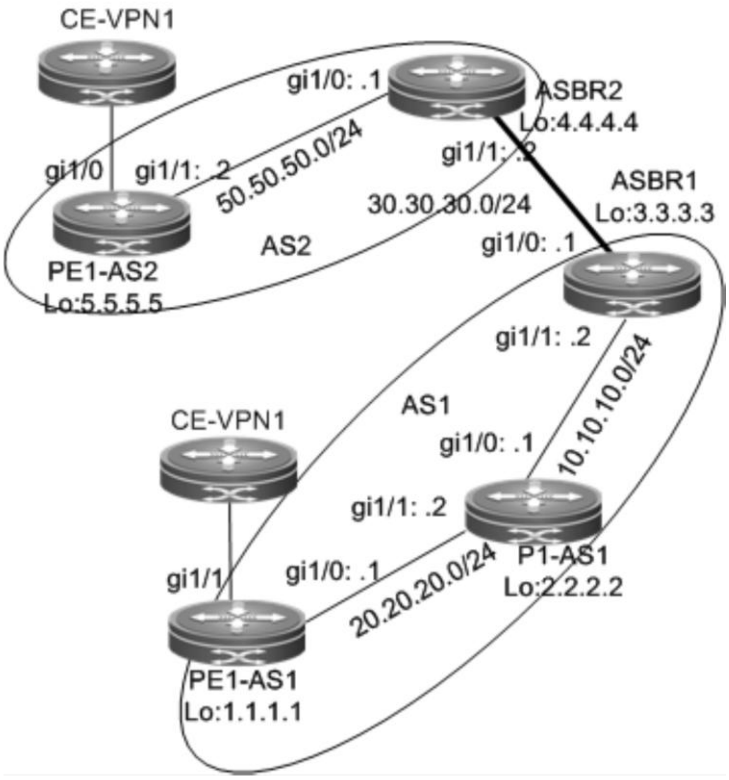
The configuration scheme of ASBR2 is similar to that of ASBR1.

Inter-AS VPN OptionC: Enabling IPv4 Label Switching Between EBGP Neighbors

Networking Requirements

One VPN user has sites at both ASs. It is required that the VPN sites in different ASs access each other.

Figure 36 OptionC: enabling IPv4 label switching between EBGP neighbors



The configuration scheme is as follows:

PE1-AS1:

# Configure the loopback interface.

```
Ruijie# configure terminal
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 1.1.1.1 255.255.255.255
```

# Configure the VRF.

The configuration procedure is similar to that of PE1-AS1 in the "Inter-AS VPN OptionB: Next Hop Unchanged" section and is not described here.

# Configure a multi-hop MP-EBGP session and disable IPv4 route exchange for the session.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 5.5.5.5 remote-as 2
Ruijie(config-router)# neighbor 5.5.5.5 update-source loopback 0
Ruijie(config-router)# neighbor 5.5.5.5 ebgp-multihop
Ruijie(config-router)# address-family ipv4
Ruijie(config-router-af)# no neighbor 5.5.5.5 activate
Ruijie(config-router-af)# exit
Ruijie(config-router)# address-family vpnv4 unicast
Ruijie(config-router-af)# neighbor 5.5.5.5 activate
Ruijie(config-router-af)# end
```

# Configure CE neighbors through EBGP.

See the configuration procedure in the "Running BGP Between PEs and CEs to Transmit Route Information" section and the related configurations in the "Intranet Configuration Examples" section.

# Configure MPLS signaling and enable MPLS on a public network interface.

```
Ruijie# configure terminal
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# exit
Ruijie(config)# interface gigabitethernet 1/1
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-gigabitethernet 1/1)# no switchport
Ruijie(config-if-gigabitethernet 1/1)# ip address 20.20.20.1
255.255.255.0
Ruijie(config-if-gigabitethernet 1/1)# label-switching
Ruijie(config-if-gigabitethernet 1/1)# mpls ip
Ruijie(config-if-gigabitethernet 1/1)# end
```

# Run OSPF on the backbone network to transmit route information.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 20.20.20.0 0.0.0.255 area 0
Ruijie(config-router)# network 1.1.1.1 0.0.0.0 area 0
Ruijie(config-router)# end
```

P1-AS1:

The configuration mainly includes the MPLS signaling protocol and IGP and is not described here. See the P1-AS1 configuration scheme in the "Inter-AS VPN OptionB: Next Hop Unchanged" section.

ASBR1:

# Configure the loopback interface.

```
Ruijie# configure terminal
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 3.3.3.3 255.255.255.255
```

# Configure ACL rules and route map rules to distribute or set labels only for routes that match the rules.

```
Ruijie# configure terminal
Ruijie(config)# ip access-list extended 101
Ruijie(config-ext-nacl)# permit ip host 1.1.1.1 any
uijie(config-ext-nacl)# exit
Ruijie(config)# ip access-list extended 102
Ruijie(config-ext-nacl)# permit ip host 5.5.5.5 any
Ruijie(config-ext-nacl)# exit
Ruijie(config)# route-map set-mpls
Ruijie(config-route-map)# match ip address 101
Ruijie(config-route-map)# set mpls-label
Ruijie(config-route-map)# exit
Ruijie(config)# route-map external-pe-route
Ruijie(config-route-map)# match ip address 102
Ruijie(config-route-map)# end
```

# Set up an EBGP session with ASBR2 and configure route map rules to distribute labels for PE routes that match the rules (the route map rules are optional and allow BGP to distribute labels for only certain routes), and configure static routes to PEs in the local AS.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 30.30.30.2 remote-as 2
Ruijie(config-router)# address-family ipv4
Ruijie(config-router-af)# neighbor 30.30.30.2 send-label
Ruijie(config-router-af)# neighbor 30.30.30.2 route-map set-mpls out
Ruijie(config-router-af)# network 1.1.1.1 mask 255.255.255.255
Ruijie(config-router-af)# end
```

# Configure MPLS to distribute label for certain BGP routes through ACL rules (the ACL rules are optional and allow you to reduce the number of unnecessary routes).

```
Ruijie# configure terminal
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# advertise-labels for bgp-routes acl 102
Ruijie(config-mpls-router)# exit
Ruijie(config)# interface gigabitethernet 1/1
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-Gigabitethernet 1/1)# no switchport
Ruijie(config-if-Gigabitethernet 1/1)# ip address 10.10.10.2 255.255.255.0
Ruijie(config-if-Gigabitethernet 1/1)# label-switching
Ruijie(config-if-Gigabitethernet 1/1)# mpls ip
Ruijie(config-if-Gigabitethernet 1/1)# end
```

# Configure a routing protocol on the backbone network to redistribute only BGP routes that match the route map rules (the route map rules are optional and allow you to reduce the number of unnecessary routes).

```
Ruijie# configure terminal
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 10.10.10.0 0.0.0.255 area 0
Ruijie(config-router)# network 3.3.3.3 0.0.0.0 area 0
Ruijie(config-router)# redisteibute bgp subnets route-map external-pe-route
Ruijie(config-router)# end
```

# Assign an IP address to the interface connected to ASBR2.

```
Ruijie(config)# interface gigabitethernet 1/0
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-GigabitEthernet 1/0)# no switchport
Ruijie(config-if-gigabitethernet 1/0)# ip address 30.30.30.1
255.255.255.0
```

# Enable label switching on an interface.

```
Ruijie(config-if-gigabitethernet 1/0)# label-switching
```

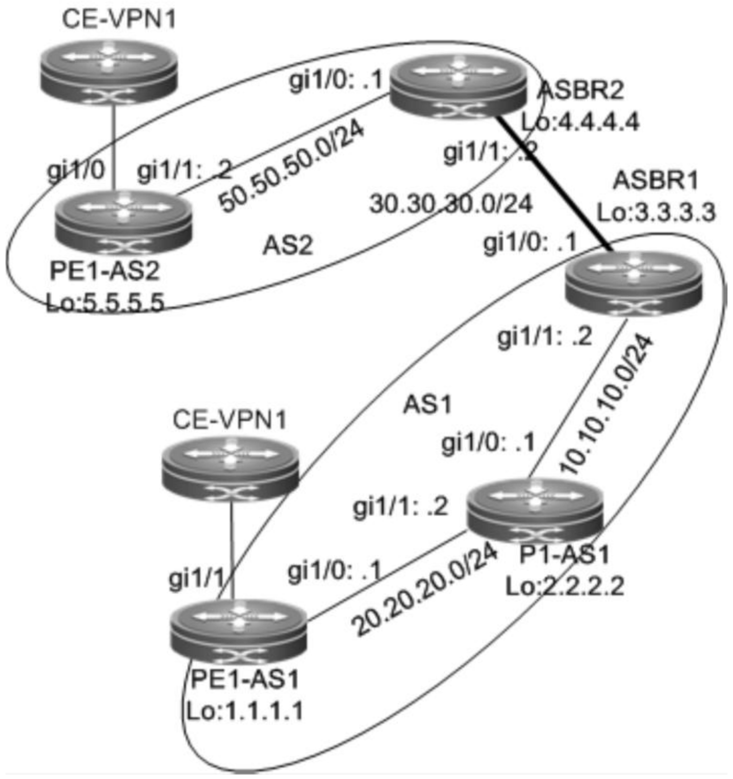
The configuration scheme of ASBR2 is similar to that of ASBR1.

Inter-AS VPN OptionC: Enabling IPv4 Label Switching Between Both EBGP and IBGP Neighbors

Networking Requirements

One VPN user has sites at both ASs. It is required that the VPN sites in different ASs access each other.

Figure 37 OptionC: enabling IPv4 label switching between both EBGP and IBGP neighbors



The configuration scheme is as follows:

PE1-AS1:

# Configure the loopback interface.

```
Ruijie# configure terminal
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 1.1.1.1 255.255.255.255
```

# Configure the VRF.

The configuration procedure is similar to that of PE1-AS1 in the "Inter-AS VPN OptionB: Next Hop Unchanged" section and is not described here.

# Configure a multi-hop MP-EBGP session and disable IPv4 route exchange for the session.

The configuration procedure is similar to that of "Inter-AS VPN OptionC: Enabling IPv4 Label Switching Between EBGP Neighbors" and is not described here.

# Set up an IBGP session with the ASBR and enable IPv4 label switching.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 3.3.3.3 remote-as 1
Ruijie(config-router)# neighbor 3.3.3.3 update-source loopback 0
Ruijie(config-router)# address-family ipv4
Ruijie(config-router-af)# neighbor 3.3.3.3 activate
Ruijie(config-router-af)# neighbor 3.3.3.3 send-label
Ruijie(config-router-af)# end
```

# Configure CE neighbors through EBGP.

See the configuration procedure in the "Running BGP Between PEs and CEs to Transmit Route Information" section and the related configurations in the "Intranet Configuration Examples" section.

# Configure MPLS signaling and enable MPLS on a public network interface.

```
Ruijie# configure terminal
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# exit
Ruijie(config)# interface gigabitethernet 1/0
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-gigabitethernet 1/0)# no switchport
Ruijie(config-if-gigabitethernet 1/0)# ip address 20.20.20.1
255.255.255.0
Ruijie(config-if-gigabitethernet 1/0)# label-switching
Ruijie(config-if-gigabitethernet 1/0)# mpls ip
Ruijie(config-if-gigabitethernet 1/0)# end
```

# Run OSPF on the backbone network to transmit route information.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 20.20.20.0 0.0.0.255 area 0
Ruijie(config-router)# network 1.1.1.1 0.0.0.0 area 0
Ruijie(config-router)# end
```

The configuration of PE1-AS2 is similar to that of PE1-AS1.

P1-AS1:

The configuration mainly includes the MPLS signaling protocol and IGP and is not described here. It is similar to "Example of Configuring Basic MPLS Functions".

ASBR1:

# Configure the loopback interface.

```
Ruijie# configure terminal
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 3.3.3.3 255.255.255.255
```

# Configure ACL rules and route map rules to distribute or set labels only for routes that match the rules.

```
Ruijie# configure terminal
Ruijie(config)# ip access-list extended 101
Ruijie(config)# permit ip host 1.1.1.1 any
Ruijie(config)# exit
Ruijie(config)# ip access-list extended 102
Ruijie(config)# permit ip host 5.5.5.5 any
Ruijie(config)# route-map internal-mpls-route permit 10
Ruijie(config-route-map)# match ip address 101
Ruijie(config-route-map)# set mpls-label
Ruijie(config-route-map)# exit
Ruijie(config)# route-map external-mpls-route permit 10
Ruijie(config-route-map)#  match ip address 102
Ruijie(config-route-map)#  set mpls-label
Ruijie(config-route-map)#  end
```

# Set up an EBGP session with the ASBR and configure route map rules to distribute labels for PE routes that match the rules (the route map rules are optional and allow BGP to distribute labels for only certain routes), and configure static routes to PEs in the local AS.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 30.30.30.2 remote-as 2
Ruijie(config-router)# neighbor 1.1.1.1 remote-as 1
Ruijie(config-router)# neighbor 1.1.1.1 update-source loopback 0
Ruijie(config-router)# address-family ipv4
Ruijie(config-router-af)# neighbor 30.30.30.2 send-label
Ruijie(config-router-af)# neighbor 30.30.30.2 route-map internal-mpls-route out
Ruijie(config-router-af)# neighbor 1.1.1.1 send-label
Ruijie(config-router-af)# neighbor 1.1.1.1 route-map external-mpls-route out
Ruijie(config-router-af)#  network 1.1.1.1 mask 255.255.255.255
Ruijie(config-router-af)# end
```

# Configure MPLS signaling and enable MPLS on an interface.

```
Ruijie# configure terminal
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# exit
Ruijie(config)# interface gigabitethernet 1/1
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-Gigabitethernet 1/1)# no switchport
Ruijie(config-if-Gigabitethernet 1/1)# ip address 10.10.10.2
255.255.255.0
Ruijie(config-if-Gigabitethernet 1/1)# label-switching
Ruijie(config-if-Gigabitethernet 1/1)# mpls ip
Ruijie(config-if-Gigabitethernet 1/1)# end
```

# Run OSPF on the backbone network to transmit route information.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 10.10.10.0 255.255.255.0 area 0
Ruijie(config-router)# network 3.3.3.3 0.0.0.0 area 0
Ruijie(config-router)# end
```

# Assign an IP address to the interface connected to ASBR2.

```
Ruijie(config)# interface gigabitethernet 1/0
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-GigabitEthernet 1/0)# no switchport
Ruijie(config-if-gigabitethernet 1/0)# ip address 30.30.30.1
255.255.255.0
```

# Enable label switching on an interface.

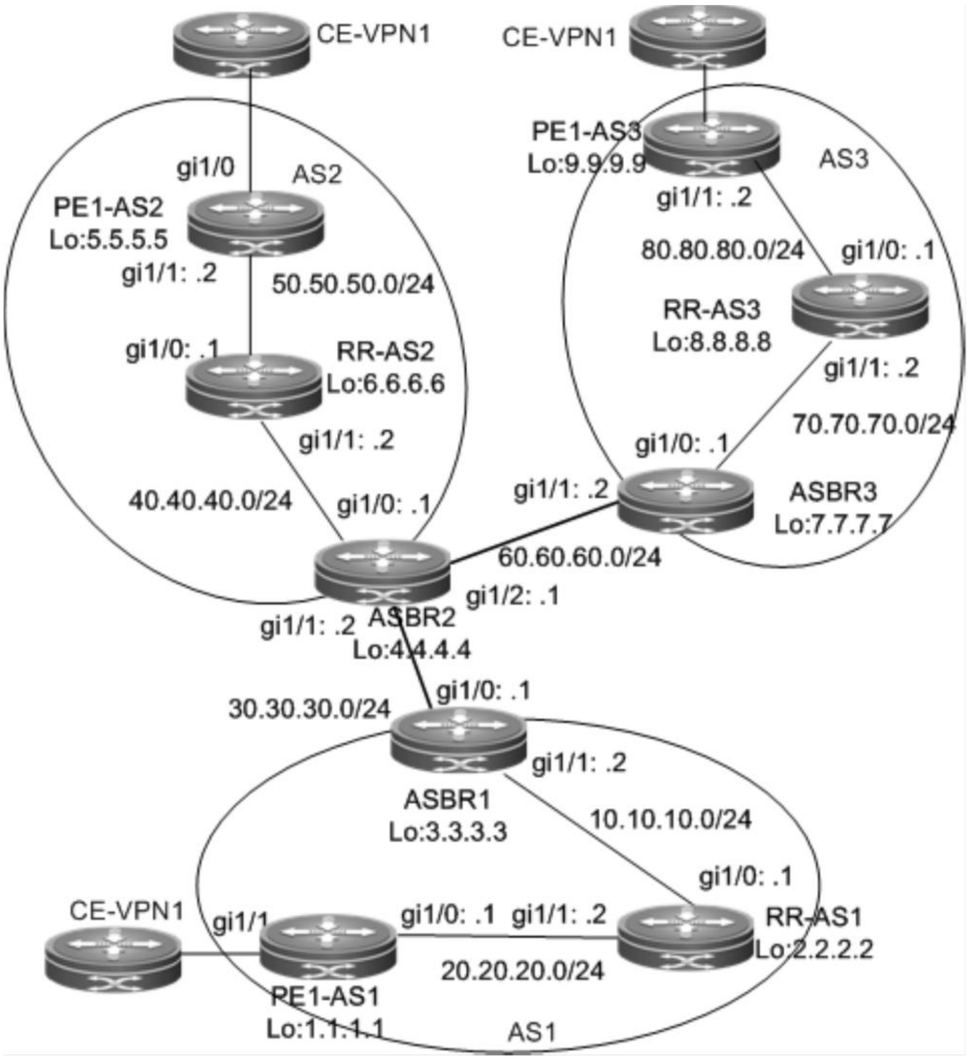
```
Ruijie(config-if-gigabitethernet 1/0)# label-switching
```

The configuration scheme of ASBR2 is similar to that of ASBR1.

Inter-AS VPN OptionC: RR Networking Scheme

In the two implementation modes of OptionC, another problem exists. If the sites of the same VPN user are located at different ASs, a common OptionC scheme requires full mesh BGP connections for the inter-AS PEs to ensure the reachability of the VPN sites. As shown in the following figure, the sites of the VPN user are located at three different ASs. If a new VPN site is added, the new site needs to set up BGP connections with the other VPN sites. This restricts the application of the common OptionC scheme. To solve the preceding expansion problem, you can add an RR to each AS in the OptionC scheme. The RRs set up multi-hop MP-EBGP connections to exchange inter-AS VPN routes. At the same time, you can set up MP-IBGP sessions between PEs and the RR in the AS.

Figure 38 Setting up multi-hop MP-EBGP sessions between RRs in the OptionC scheme



The configuration scheme is as follows:

PE1-AS1:

# Configure the loopback interface.

```
Ruijie# configure terminal
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 1.1.1.1 255.255.255.255
```

# Configure the VRF.

The configuration procedure is similar to that of PE1-AS1 in the "Inter-AS VPN OptionB: Next Hop Unchanged" section and is not described here.

# Set up an MP-IBGP session with the RR and enable label distribution for IPv4 routes.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 2.2.2.2 remote-as 1
Ruijie(config-router)# neighbor 2.2.2.2 update-source loopback 0
Ruijie(config-router)# address-family vpnv4 unicast
Ruijie(config-router-af)# neighbor 2.2.2.2 activate
Ruijie(config-router-af)# exit
Ruijie(config-router)# address-family ipv4
Ruijie(config-router-af)# neighbor 2.2.2.2 activate
Ruijie(config-router-af)# neighbor 2.2.2.2 send-label
Ruijie(config-router-af)# end
```

# Configure CE neighbors through EBGP.

See the configuration procedure in the "Running BGP Between PEs and CEs to Transmit Route Information" section and the related configurations in the "Intranet Configuration Examples" section.

The configurations of PE1-AS2 and PE1-AS3 are similar to that of PE1-AS1.

RR-AS1

# Configure the loopback interface.

```
Ruijie# configure terminal
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 2.2.2.2 255.255.255.255
```

# Set up an MP-IBGP session with the PE, specify the PE as the RR client, and enable label distribution for IPv4 routes.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 1.1.1.1 remote-as 1
Ruijie(config-router)# neighbor 1.1.1.1 update-source loopback 0
Ruijie(config-router)# address-family vpnv4 unicast
Ruijie(config-router-af)# neighbor 1.1.1.1 activate
Ruijie(config-router-af)# neighbor 1.1.1.1 route-reflector-client
Ruijie(config-router-af)# exit
Ruijie(config-router)# address-family ipv4
Ruijie(config-router-af)# neighbor 1.1.1.1 activate
Ruijie(config-router-af)# neighbor 1.1.1.1 send-label
Ruijie(config-router-af)# neighbor 1.1.1.1 route-reflector-client
Ruijie(config-router-af)# end
```

# Set up a multi-hop MP-EBGP session with the RR, do not change the next hop of VPN routes exchanged with the RR, and disable the IPv4 route exchange with the RR.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 6.6.6.6 remote-as 2
Ruijie(config-router)# neighbor 6.6.6.6 update-source loopback 0
Ruijie(config-router)# neighbor 6.6.6.6 ebgp-multihop
Ruijie(config-router)# neighbor 8.8.8.8 remote-as 3
Ruijie(config-router)# neighbor 8.8.8.8 update-source loopback 0
Ruijie(config-router)# neighbor 8.8.8.8 ebgp-multihop
Ruijie(config-router)# address-family ipv4
Ruijie(config-router-af)# no neighbor 6.6.6.6 activate
Ruijie(config-router-af)# no neighbor 8.8.8.8 activate
Ruijie(config-router-af)# exit-address-family
Ruijie(config-router)# address-family vpnv4 unicast
Ruijie(config-router-af)# neighbor 6.6.6.6 activate
Ruijie(config-router-af)# neighbor 6.6.6.6 next-hop-unchanged
Ruijie(config-router-af)# neighbor 8.8.8.8 activate
Ruijie(config-router-af)# neighbor 8.8.8.8 next-hop-unchanged
Ruijie(config-router-af)# end
```

# Set up an IBGP session with the ASBR and enable IPv4 label switching.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 3.3.3.3 remote-as 1
Ruijie(config-router)# neighbor 3.3.3.3 update-source loopback 0
Ruijie(config-router)# address-family ipv4
Ruijie(config-router-af)# neighbor 3.3.3.3 activate
Ruijie(config-router-af)# neighbor 3.3.3.3 send-label
Ruijie(config-router-af)# end
```

# Configure MPLS.

```
Ruijie# configure terminal
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# exit
Ruijie(config)# interface gigabitethernet 1/1
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-gigabitethernet 1/1)# no switchport
Ruijie(config-if-gigabitethernet 1/1)# ip address 20.20.20.2
255.255.255.0
Ruijie(config-if-gigabitethernet 1/1)# label-switching
Ruijie(config-if-gigabitethernet 1/1)# mpls ip
Ruijie(config-if-gigabitethernet 1/1)# exit
Ruijie(config)# interface gigabitethernet 1/0
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-gigabitethernet 1/0)# no switchport
Ruijie(config-if-gigabitethernet 1/0)# ip address 10.10.10.1
255.255.255.0
Ruijie(config-if-gigabitethernet 1/0)# label-switching
Ruijie(config-if-gigabitethernet 1/0)# mpls ip
Ruijie(config-if-gigabitethernet 1/0)# end
```

# Run OSPF on the backbone network to transmit route information.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 20.20.20.0 0.0.0.255 area 0
Ruijie(config-router)# network 2.2.2.2 0.0.0.0 area 0
Ruijie(config-router)# network 10.10.10.0 0.0.0.255 area 0
Ruijie(config-router)# end
```

The procedures of RR-AS2 and RR-AS3 are similar to the preceding procedure.

ASBR1:

# Configure the loopback interface.

```
Ruijie# configure terminal
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 3.3.3.3 255.255.255.255
```

# Configure ACL rules and route map rules.

```
Ruijie# configure terminal
Ruijie(config)# ip access-list extended 101
Ruijie(config-ext-nacl)# permit ip host 1.1.1.1 any
Ruijie(config-ext-nacl)# exit
Ruijie(config)# ip access-list extended 102
Ruijie(config-ext-nacl)# permit ip host 5.5.5.5 any
Ruijie(config-ext-nacl)# permit ip host 9.9.9.9 any
Ruijie(config-ext-nacl)# exit
Ruijie(config)# route-map internal-mpls-route permit 10
Ruijie(config-route-map)# match ip address 101
Ruijie(config-route-map)# set mpls-label
Ruijie(config-route-map)# exit
Ruijie(config)# route-map external-mpls-route permit 10
Ruijie(config-route-map)# match ip address 102
Ruijie(config-route-map)# set mpls-label
Ruijie(config-route-map)# end
```

# Set up an EBGP session with the ASBR, enable label distribution for IPv4 routes, and configure route map rules to distribute labels for PE routes that match the rules (the route map rules are optional and allow the BGP to distribute labels for only certain routes). Set up an IBGP session with the RR, enable label distribution for IPv4 routes, and configure route map rules to distribute labels for inter-AS PE routes that match the rules. Configure static routes to the PEs in the local AS.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 30.30.30.2 remote-as 2
Ruijie(config-router)# neighbor 2.2.2.2 remote-as 1
Ruijie(config-router)# neighbor 2.2.2.2 update-source loopback 0
Ruijie(config-router)# address-family ipv4
Ruijie(config-router-af)# neighbor 30.30.30.2 send-label
Ruijie(config-router-af)# neighbor 30.30.30.2 route-map internal-mpls-route out
Ruijie(config-router-af)# neighbor 2.2.2.2 send-label
Ruijie(config-router-af)# neighbor 2.2.2.2 route-map external-mpls-route out
Ruijie(config-router-af)# network 1.1.1.1 mask 255.255.255.255
Ruijie(config-router-af)# end
```

# Configure MPLS signaling and enable MPLS on an interface.

```
Ruijie# configure terminal
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# exit
Ruijie(config)# interface gigabitethernet 1/1
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-Gigabitethernet 1/1)# no switchport
Ruijie(config-if-Gigabitethernet 1/1)#ip address 10.10.10.2
255.255.255.0
Ruijie(config-if-Gigabitethernet 1/1)# label-switching
Ruijie(config-if-Gigabitethernet 1/1)# mpls ip
Ruijie(config-if-Gigabitethernet 1/1)# end
```

# Run OSPF on the backbone network to transmit route information.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 10.10.10.0 0.0.0.255 area 0
Ruijie(config-router)# network 3.3.3.3 0.0.0.0 area 0
Ruijie(config-router)# end
```

# Assign an IP address to the interface connected to ASBR2.

```
Ruijie(config)# interface gigabitethernet 1/0
```

# Use the **no switchport** command to switch the port mode on the switch series to the Routed Port mode. This command is not applicable to routers.

```
Ruijie(config-if-GigabitEthernet 1/0)# no switchport
Ruijie(config-if-gigabitethernet 1/0)# ip address 30.30.30.1
255.255.255.0
```

# Enable label switching on an interface.

```
Ruijie(config-if-gigabitethernet 1/0)# label-switching
```

The configuration schemes of ASBR2 and ASBR3 are similar to that of ASBR1.

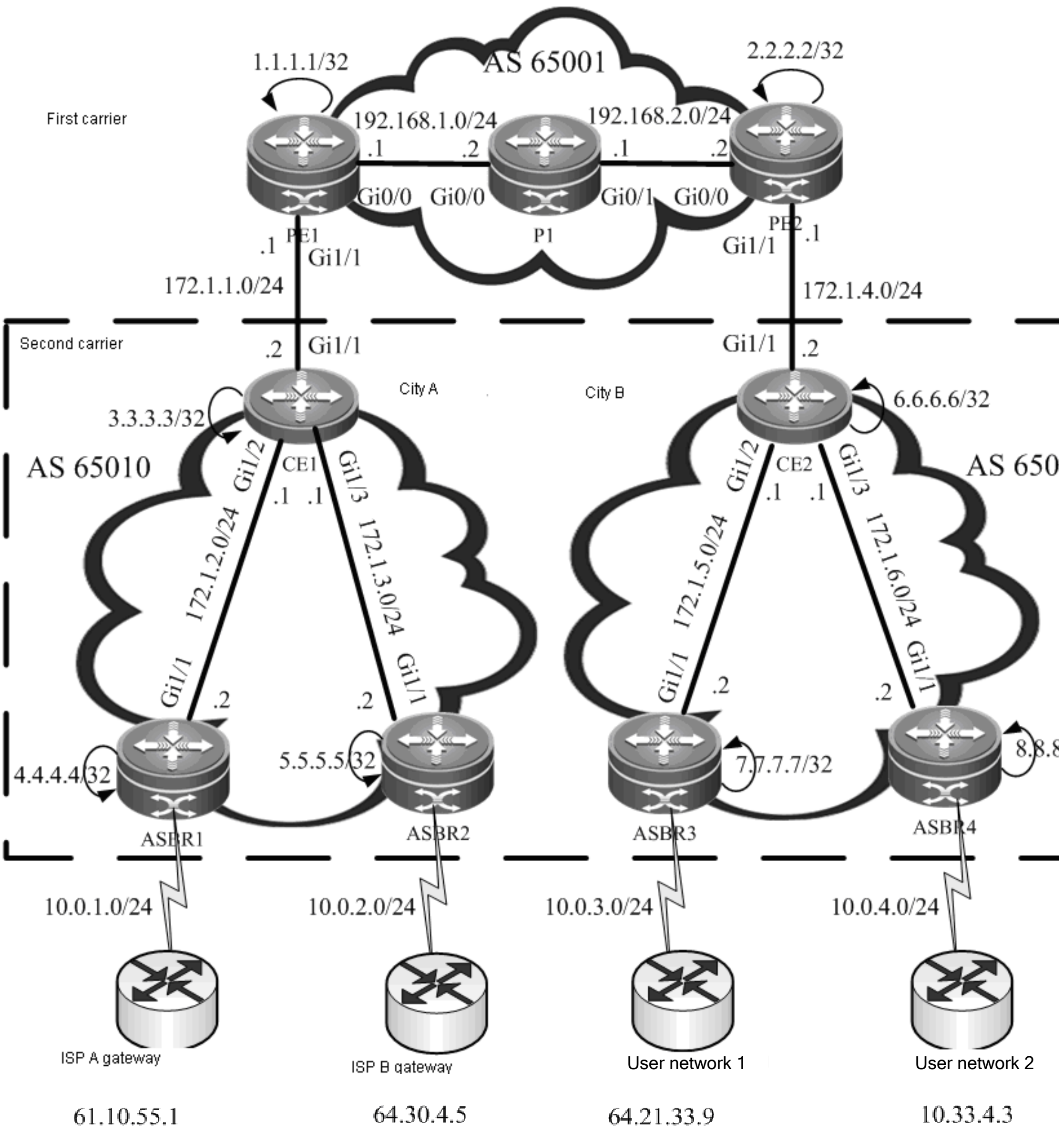
CSC: The Second Carrier Provides Internet Services Based on IP Core

Networking Requirements

A carrier owns an intranet in City A, and this network has the BGP gateways to ISP A and ISP B. This carrier utilizes its intranet to provide Internet services for users in City A. Currently, this carrier expects to expand services to City B, and therefore leases MPLS VPN services from a VPN carrier in the hope of connecting the sites of two cities via VPN, so that users in City B can access the Internet through the existing Internet gateways. The internal routes are exchanged via IGP (OSPF), and the external routes are exchanged via BGP.

Networking Topology

Figure 39 Network topology of scenario I



Configuration Tips

- Configuring basic BGP/MPLS IP VPN functions for the first carrier
- Enabling the CSC function
- Configuring the second carrier

- Configuring user access

Configuration Steps

- Configuring basic BGP/MPLS IP VPN functions for the first carrier

# Configure an MPLS network: PE1 is used as an example. The configurations of P1 and PE2 are similar.

## Configure a loopback interface.

```
Ruijie(config)# interface Loopback 0
Ruijie(config-if)# ip address 1.1.1.1 255.255.255.255
Ruijie(config-if)# exit
```

## Globally enable MPLS and LDP.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp rouer-id interface Loopback 0
Ruijie(config-mpls-router)# exit
```

## Enable MPLS and LDP on the interface.

```
Ruijie(config)# interface gigabitEthernet 0/0
```

# In case of a switch, configure the interface to a RoutedPort interface (not applicable to a router).

```
Ruijie(config-if)# no switchport
```

# In case of a router, enable fast forwarding on the interface (not applicable to a switch).

```
Ruijie(config-if)# ip ref
Ruijie(config-if)# ip address 192.168.1.1 255.255.255.0
Ruijie(config-if)# label-switching
Ruijie(config-if)# mpls ip
Ruijie(config-if)# no shutdown
Ruijie(config-if)# exit
```

## Configure IGP (OSPF).

```
Ruijie(config)# router ospf 1
Ruijie(config-router)# network 1.1.1.1 0.0.0.0 area 0
Ruijie(config-router)# network 192.168.1.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure an MP-IBGP neighbor: PE1 is used as an example. The configurations of PE2 are similar.

```
Ruijie(config)# router bgp 65001
Ruijie(config-router)# neighbor 2.2.2.2 remote-as 65001
Ruijie(config-router)# neighbor 2.2.2.2 update-source Loopback 0
Ruijie(config-router)# address-family vpnv4
Ruijie(config-router-af)# neighbor 2.2.2.2 activate
Ruijie(config-router-af)# neighbor 2.2.2.2 send-community both
```

# Configure a VRF: PE1 is used as an example. The configurations of PE2 are similar.

```
Ruijie(config)# ip vrf vpn1
Ruijie(config-vrf)# rd 65001:20
Ruijie(config-vrf)# route-target both 65001:20
Ruijie(config-vrf)# alloc-label per-route
Ruijie(config-vrf)# exit
Ruijie(config)# interface loopback 1
Ruijie(config-if)# ip vrf forwarding vpn1
Ruijie(config-if)# ip address 10.1.1.1 255.255.255.255
Ruijie(config-if)# no shutdown
Ruijie(config-if)# exit
Ruijie(config)# interface gigabitEthernet 1/1
```

# In case of a switch, configure the interface to a RoutedPort interface (not applicable to a router).

```
Ruijie(config-if)# no switchport
```

# In case of a router, enable fast forwarding on the interface (not applicable to a switch).

```
Ruijie(config-if)# ip ref
Ruijie(config-if)# ip vrf forwarding vpn1
Ruijie(config-if)# ip address 172.1.1.1 255.255.255.0
Ruijie(config-if)# no shutdown
```

# Configure a CE to connect to a PE: CE1 is used as an example. The configurations of CE2 are similar.

```
Ruijie(config)# interface gigabitEthernet 1/1
```

# In case of a switch, configure the interface to a RoutedPort interface (not applicable to a router).

```
Ruijie(config-if)# no switchport
```

# In case of a router, enable fast forwarding on the interface (not applicable to a switch).

```
Ruijie(config-if)# ip ref
Ruijie(config-if)# ip address 172.1.1.2 255.255.255.0
Ruijie(config-if)# no shutdown
```

# Configure route exchange between PEs and CEs: Route exchange between PE1 and CE1 is used as an example. The configurations of route exchange between PE2 and CE2 are similar.

## First, configure PE1.

```
Ruijie(config)# router ospf 100 vrf vpn1
Ruijie(config-router)# network 172.1.1.0 0.0.0.255 area 0
Ruijie(config-router)# redistribute bgp 65001 subnets
Ruijie(config-router)# exit
Ruijie(config)# router bgp 65001
Ruijie(config-router)# address-family ipv4 vrf vpn1
Ruijie(config-router-af)# redistribute ospf 100 vrf vpn1
Ruijie(config-router-af)# exit
Ruijie(config-router)# exit
```

## Then, configure CE1.

```
Ruijie(config)# router ospf 1
Ruijie(config-router)# network 172.1.1.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

- Enabling the CSC function

# Enable CSC on the PE: PE1 is used as an example. The configurations of PE2 are similar.

```
Ruijie(config)# mpls router ldp vpn1
Ruijie(config-mpls-router)# ldp rouer-id interface Loopback 1
Ruijie(config-mpls-router)# advertise-labels for bgp-routes
Ruijie(config-mpls-router)# exit
Ruijie(config)# interface gigabitEthernet 1/1
```

# In case of a switch, configure the interface to a RoutedPort interface (not applicable to a router).

```
Ruijie(config-if)# no switchport
```

# In case of a router, enable fast forwarding on the interface (not applicable to a switch).

```
Ruijie(config-if)# ip ref
Ruijie(config-if)# label-switching
Ruijie(config-if)# mpls ip
```

# Enable MPLS and LDP on the CE: CE1 is used as an example. The configurations of CE2 are similar.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp rouer-id interface Loopback 0
Ruijie(config-mpls-router)# exit
Ruijie(config)# interface gigabitEthernet 1/1
```

# In case of a switch, configure the interface to a RoutedPort interface (not applicable to a router).

```
Ruijie(config-if)# no switchport
```

# In case of a router, enable fast forwarding on the interface (not applicable to a switch).

```
Ruijie(config-if)# ip ref
Ruijie(config-if)# label-switching
Ruijie(config-if)# mpls ip
```

- Configuring the second carrier

# Configure the interface and IGP: CE1 is used as an example. The configurations of CE2, ASBR1, ASBR2, ASBR3 and ASBR4 are similar.

```
Ruijie(config)# interface gigabitEthernet 1/2
```

# In case of a switch, configure the interface to a RoutedPort interface (not applicable to a router).

```
Ruijie(config-if)# no switchport
```

# In case of a router, enable fast forwarding on the interface (not applicable to a switch).

```
Ruijie(config-if)# ip ref
Ruijie(config-if)# ip address 172.1.2.1 255.255.255.0
Ruijie(config-if)# no shutdown
Ruijie(config-if)# exit
Ruijie(config)# interface gigabitEthernet 1/3
```

# In case of a switch, configure the interface to a RoutedPort interface (not applicable to a router).

```
Ruijie(config-if)# no switchport
```

# In case of a router, enable fast forwarding on the interface (not applicable to a switch).

```
Ruijie(config-if)# ip ref
Ruijie(config-if)# ip address 172.1.3.1 255.255.255.0
Ruijie(config-if)# no shutdown
Ruijie(config-if)# exit
Ruijie(config)# interface Loopback 0
Ruijie(config-if)# ip address 3.3.3.3 255.255.255.255
Ruijie(config-if)# exit
Ruijie(config)# router ospf 1
Ruijie(config-router)# network 3.3.3.3 0.0.0.0 area 0
Ruijie(config-router)# network 172.1.2.0 0.0.0.255 area 0
Ruijie(config-router)# network 172.1.3.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# On the ASBR, configure the CE as a BGP neighbor: ASBR1 is used as an example. The configurations of ASBR2, ASBR3 and ASBR4 are similar.

```
Ruijie(config)# router bgp 65010
Ruijie(config-router)# neighbor 3.3.3.3 remote-as 65010
Ruijie(config-router)# neighbor 3.3.3.3 update-source Loopback 0
Ruijie(config-router)# neighbor 3.3.3.3 next-hop-self
```

# On the CE, configure the ASBR and a peer CE as the RR client and enable resolution of the next hop of a BGP route to an LSP tunnel: CE1 is used as an example. The configurations of CE2 are similar.

```
Ruijie(config)# router bgp 65010
Ruijie(config-router)# neighbor 4.4.4.4 remote-as 65010
Ruijie(config-router)# neighbor 4.4.4.4 update-source Loopback 0
Ruijie(config-router)# neighbor 4.4.4.4 route-reflector-client
Ruijie(config-router)# neighbor 5.5.5.5 remote-as 65010
Ruijie(config-router)# neighbor 5.5.5.5 update-source Loopback 0
Ruijie(config-router)# neighbor 5.5.5.5 route-reflector-client
Ruijie(config-router)# neighbor 6.6.6.6 remote-as 65010
Ruijie(config-router)# neighbor 6.6.6.6 update-source Loopback 0
Ruijie(config-router)# neighbor 6.6.6.6 route-reflector-client
Ruijie(config-router)# exit
Ruijie(config)# recursive-route lookup lsp
```

- Configuring user access

# It is assumed that user network 1 is connected to ASBR3. The configurations of other external networks (user network and Internet gateway) are similar.

## On ASBR3, use the following command.

```
Ruijie(config)# interface gigabitEthernet 1/2
```

# In case of a switch, configure the interface to a RoutedPort interface (not applicable to a router).

```
Ruijie(config-if)# no switchport
```

# In case of a router, enable fast forwarding on the interface (not applicable to a switch).

```
Ruijie(config-if)# ip ref
Ruijie(config-if)# ip address 10.0.3.1 255.255.255.0
Ruijie(config-if)# no shutdown
Ruijie(config-if)# exit
Ruijie(config)# router bgp 65010
Ruijie(config-router)# neighbor 10.0.3.2 remote-as 100
Ruijie(config-router)# exit
```

## On the edge router of user network 1, use the following command.

```
Ruijie(config)# interface gigabitEthernet 0/0
```

# In case of a switch, configure the interface to a RoutedPort interface (not applicable to a router).

```
Ruijie(config-if)# no switchport
```

# In case of a router, enable fast forwarding on the interface (not applicable to a switch).

```
Ruijie(config-if)# ip ref
Ruijie(config-if)# ip address 10.0.3.2 255.255.255.0
Ruijie(config-if)# no shutdown
Ruijie(config-if)# exit
Ruijie(config)# interface gigabitEthernet 0/1
```

# In case of a switch, configure the interface to a RoutedPort interface (not applicable to a router).

```
Ruijie(config-if)# no switchport
```

# In case of a router, enable fast forwarding on the interface (not applicable to a switch).

```
Ruijie(config-if)# ip ref
Ruijie(config-if)# ip address 64.21.33.9 255.255.255.0
Ruijie(config-if)# no shutdown
Ruijie(config-if)# exit
Ruijie(config)# router bgp 100
Ruijie(config-router)# neighbor 10.0.3.1 remote-as 65010
Ruijie(config-router)# network 64.21.33.0 mask 255.255.255.0
```

Verification

# Display VRF routes and labels on the PE: PE1 is used as an example. The verification of PE2 is similar.

// In the VRF routing table of the PE, there are only internal routes of the second carrier. There is no external route (for example, 64.30.4.0/24).

```
Ruijie# show ip route vrf vpn1
Routing Table: vpn1

Codes:  C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
O    3.3.3.3/32 [110/11] via 172.1.1.2, 00:00:07, gigabitEthernet 1/1
C    172.1.1.0/24 is directly connected, gigabitEthernet 1/1
C    172.1.1.1/32 is local host.
O    172.1.2.0/24 [110/12] via 172.1.1.2, 00:00:07, gigabitEthernet 1/1
B    172.1.4.0/24 [200/0] via 2.2.2.2, 00:00:30
......
Ruijie# show mpls ldp bindings vrf vpn1
VRF vpn1(id 1)
    lib entry: 3.3.3.3/32
        local binding:  to lsr:   172.1.1.2:0, label: 1025
        remote binding: from lsr: 172.1.1.2:0, label: imp-null
    lib entry: 172.1.1.0/24
        local binding:  to lsr:   172.1.1.2:0, label: imp-null
        remote binding: from lsr: 172.1.1.2:0, label: imp-null
    lib entry 172.1.2.0/24
        local binding:  to lsr:   172.1.1.2:0, label: 1026
        remote binding: from lsr: 172.1.1.2:0, label: 1024
    ......
```

# On the ASBR and user network, display the routing table.

// On the ASBR, there are both external routes and internal routes (using ASBR3 as the example).

```
Ruijie# show ip route
Codes:  C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
......
O    3.3.3.3/24 [110/12] via 172.1.5.1, 00:00:30, gigabitEthernet 1/1
B    61.10.55.0/24 [200/0] via 3.3.3.3, 00:00:40
B    64.21.33.0/24 [200/0] via 10.0.3.2, 00:00:31
```

```
......
```

// In the user network, there are external routes (using the edge device of user network 1 as the example).

```
Ruijie# show ip route
Ruijie# show ip route
Codes:  C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
......
B    61.10.55.0/24 [200/0] via 10.0.3.1, 00:00:40
C    64.21.33.0/24 is directly connected, gigabitEthernet 0/1
C    64.21.33.9/32 is local host.
......
```

# Verify that the external networks are interconnected.

// On the edge device of user network 1, use the following command.

```
Ruijie# ping 61.10.55.1 source 64.21.33.9
Sending 5, 100-byte ICMP Echoes to 61.10.55.1, timeout is 2 seconds:
Packet sent with a source address of 64.21.33.9
  < press Ctrl+C to break >
!!!!!
```

CSC: The Second Carrier Provides Internet Services Based on MPLS Core

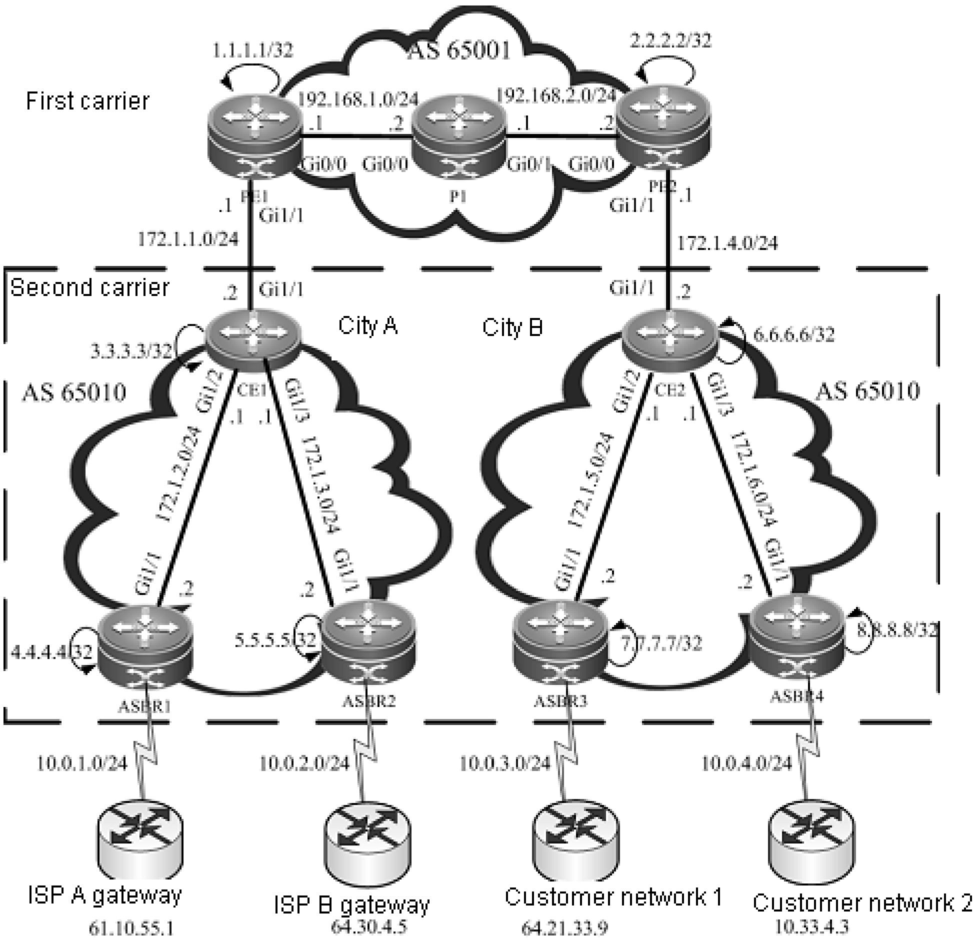
Networking Requirements

A carrier provides Internet services for users in City A. Considering that it may need to provide MPLS services for users in the future, this carrier has deployed MPLS on its backbone network. Now this carrier intends to expand its service to City B, and has built an MPLS network in City B. To interconnect the core networks in the two cities, this carrier leases the VPN service from another MPLS VPN service provider. Therefore, this carrier has become a second carrier, while the MPLS VPN service provider is the first carrier.

The first carrier PE and second carrier CE will exchange (internal) routes via BGP. The second carrier will directly establish BGP neighbors between ASBRs to exchange external routes. The traffic will flow from the external network into the second carrier network and be forwarded on the tunnel until the traffic leaves the second carrier network.

Networking Topology

Figure 40 Network topology of scenario II



Configuration Tips

- Configuring basic BGP/MPLS IP VPN functions for the first carrier
- Enabling the CSC function
- Configuring the second carrier
- Configuring user access

Configuration Steps

- Configuring basic BGP/MPLS IP VPN functions for the first carrier

The configuration steps are similar to those in Scenario I. The difference is that routes are exchanged between PEs and CEs. Only configurations of route exchange between PEs and CEs will be shown below. For other configurations, see "Configuring basic BGP/MPLS IP VPN functions" in the example of "The Second Carrier Provides Internet Services Based on IP Core".

# Configure route exchange between PEs and CEs.

## First, configure the PE (using PE1 as an example).

```
Ruijie(config)# router bgp 65001
Ruijie(config-router)# address-family ipv4 vrf vpn1
Ruijie(config-router-af)# neighbor 172.1.1.2 remote-as 65010
Ruijie(config-router-af)# neighbor 172.1.1.2 as-override
Ruijie(config-router-af)# exit
Ruijie(config-router)# exit
```

## Then, configure the CE (using CE1 as an example).

```
Ruijie(config)# router bgp 65010
Ruijie(config-router)# neighbor 172.1.1.2 remote-as 65001
Ruijie(config-router)# redistribute ospf 1
Ruijie(config-router)# exit
Ruijie(config)# router ospf 1
Ruijie(config-router)# redistribute bgp 65010 subnets
Ruijie(config-router)# exit
```

- Enabling the CSC function

# Enable CSC on the PE: PE1 is used as an example. The configurations of PE2 are similar.

```
Ruijie(config)# interface gigabitEthernet 1/1
```

# In case of a switch, configure the interface to a RoutedPort interface (not applicable to a router).

```
Ruijie(config-if)# no switchport
```

# In case of a router, enable fast forwarding on the interface (not applicable to a switch).

```
Ruijie(config-if)# ip ref
Ruijie(config-if)# ip vrf forwarding vpn1
Ruijie(config-if)# ip address 172.1.1.1 255.255.255.0
Ruijie(config-if)# no shutdown
Ruijie(config-if)# exit
Ruijie(config)# router bgp 65001
Ruijie(config-router)# address-family ipv4 vrf vpn1
Ruijie(config-router-af)# neighbor 172.1.1.2 send-label
Ruijie(config-router-af)# exit
Ruijie(config-router)# exit
```

# Enable MPLS and BGP label distribution on the CE.

```
Ruijie(config)# interface gigabitEthernet 1/1
```

# In case of a switch, configure the interface to a RoutedPort interface (not applicable to a router).

```
Ruijie(config-if)# no switchport
```

# In case of a router, enable fast forwarding on the interface (not applicable to a switch).

```
Ruijie(config-if)# ip ref
Ruijie(config-if)# label-switching
Ruijie(config-if)# ip address 172.1.1.2 255.255.255.0
Ruijie(config-if)# no shutdown
Ruijie(config-if)# exit
Ruijie(config)# router bgp 65010
Ruijie(config-router)# neighbor 172.1.1.1 send-label
Ruijie(config-router)# exit
```

- Configuring the second carrier

# Configure an MPLS network: See "Configuring an MPLS network" in the example of "The Second Carrier Provides Internet Services Based on IP Core". Configuration objects are CE1, CE2 and ASBRs (1 to 4).

Note

You need to enable LDP on the CSC-CE in order to set up sessions with other intra-site devices to build an MPLS network. If the CSC-CE and CSC-PE use BGP to exchange routes, you must use the **advertise-labels for bgp-routes** command on the CSC-CE to allow LDP to distribute labels for BGP routes.

# Configure a BGP neighbor: Establish a BGP neighbor relation between two ASBRs.

## Configure ASBR2 as the BGP neighbor on ASBR1. The configurations of other ASBRs are similar.

```
Ruijie(config)# router bgp 65010
Ruijie(config-router)# neighbor 5.5.5.5 remote-as 65010
Ruijie(config-router)# neighbor 5.5.5.5 update-source Loopback 0
Ruijie(config-router)# neighbor 5.5.5.5 next-hop-self
Ruijie(config-router)# exit
Ruijie(config)# recursive-route lookup lsp
```

- Configuring user access

See "Configuring user access" in the example of "The Second Carrier Provides Internet Services Based on IP Core".

Verification

# Display VRF routes and labels on the first carrier PE: PE1 is used as an example. The verification of PE2 is similar.

// In the VRF routing table of PE1, there are only internal routes of the second carrier. There is no external routes (for example, 64.30.4.0/24).

```
Ruijie# show ip route vrf vpn1
Routing Table: vpn1

Codes:  C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
B    3.3.3.3/32 [200/0] via 172.1.1.2, 00:00:07
C    172.1.1.0/24 is directly connected, gigabitEthernet 1/1
C    172.1.1.1/32 is local host.
B    172.1.2.0/24 [200/0] via 172.1.1.2, 00:00:07
B    172.1.4.0/24 [200/0] via 2.2.2.2, 00:00:30
......
Ruijie# show bgp vpnv4 unicast vrf vpn1 labels
```

```
BGP table version is 1, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
            S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

  Network         Next Hop       In Label/Out Label
Route Distinguisher: 65001:20 (Default for VRF vpn1)
*> 3.3.3.3/32      172.1.1.2       2048/1024
*> 172.1.2.0/24    172.1.1.2       2049/1025
*>i6.6.6.6/32      2.2.2.2         2050/2112
......
```

# On the ASBR and user network, display the routing table.

// On the ASBR (using ASBR3 as an example), use the following command.

```
Ruijie# show ip route
Codes:  C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
......
B    61.10.55.0/24 [200/0] via 4.4.4.4, 00:00:40
B    64.21.33.0/24 [200/0] via 10.0.3.2, 00:00:31
......
```

// In the user network, use the edge device of user network 1 as an example.

```
Ruijie# show ip route
Codes:  C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
......
B    61.10.55.0/24 [200/0] via 10.0.3.1, 00:00:40
C    64.21.33.0/24 is directly connected, gigabitEthernet 0/1
C    64.21.33.9/32 is local host.
......
```

# Verify that the external networks are interconnected.

// On the edge device of user network 1, use the following command.

```
Ruijie# ping 61.10.55.1 source 64.21.33.9
Sending 5, 100-byte ICMP Echoes to 61.10.55.1, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 10/20/40 ms
```

CSC: The Second Carrier Provides VPN Services Based on MPLS Core

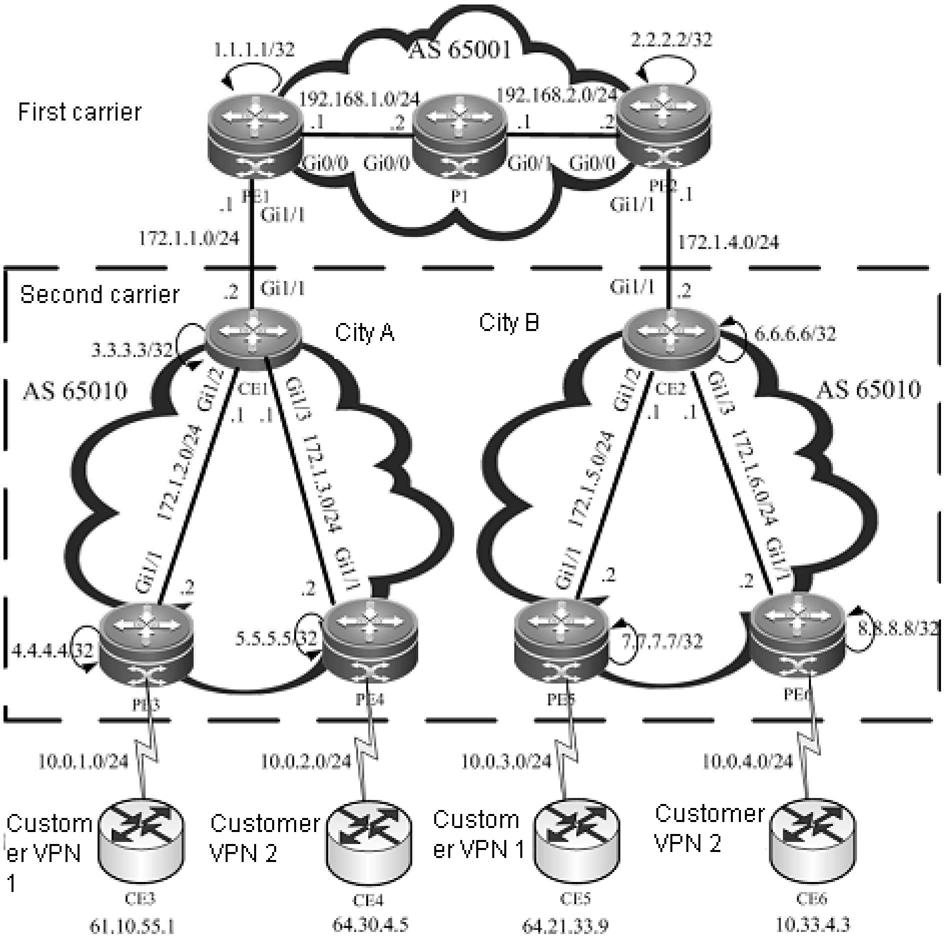
Networking Requirements

A carrier owns an MPLS core network in City A and provides MPLS VPN services for users in this city. Now this carrier intends to expand the service to City B, and has built an MPLS core network in City B. In order to interconnect the core networks in these two cities, this carrier leases the VPN service from another MPLS VPN service provider, thus forming the CSC networking model.

The first carrier PE and second carrier CE will exchange (internal) routes via BGP. An MP-IBGP neighbor relation is established between second carrier PEs to exchange user VPN routes. OSPF is deployed between the second carrier PE and the user VPN CE to exchange routes.

Networking Topology

Figure 41 MPLS core second VPN provider



Configuration Tips

- Configuring basic BGP/MPLS IP VPN functions for the first carrier
- Enabling the CSC function
- Configuring the second carrier
- Configuring user access

Configuration Steps

- Configuring basic BGP/MPLS IP VPN functions for the first carrier

See "Configuring basic BGP/MPLS IP VPN functions" in the example of "The Second Carrier Provides Internet Services Based on MPLS Core".

- Enabling the CSC function

See "Enabling the CSC function" in the example of "The Second Carrier Provides Internet Services Based on MPLS Core".

- Configuring the second carrier

Configure an MPLS network: See "Configuring an MPLS network" in the example of "The Second Carrier Provides Internet Services Based on MPLS Core". Configuration objects are CE1, CE2 and PEs (3 to 6).

Note

You need to enable LDP on the CSC-CE in order to set up sessions with other intra-site devices to build an MPLS network. If the CSC-CE and CSC-PE use BGP to exchange routes, you must run **advertise-labels for bgp-routes** on the CSC-CE to allow LDP to distribute labels for BGP routes.

Configure an MP-IBGP neighbor: See "Configuring an MP-IBGP neighbor" in the example of "The Second Carrier Provides Internet Services Based on IP Core". Configure the MP-IBGP neighbor relations between PE3, PE4, PE5 and PE6.

- Configuring user access

The configurations include configuring a VRF, configuring route exchange between PEs and CEs, and so on. These configurations are the same as those of the BGP/MPLS IP VPN. It is assumed that CE3 is connected to PE3.

# On PE3, use the following commands.

```
Ruijie(config)# ip vrf customer_vpn1
Ruijie(config-vrf)# rd 65010:1
Ruijie(config-vrf)# route-target both 65010:1
Ruijie(config-vrf)# exit
Ruijie(config)# interface gigabitEthernet 1/2
```

# In case of a switch, configure the interface to a RoutedPort interface (not applicable to a router).

```
Ruijie(config-if)# no switchport
```

# In case of a router, enable MPLS fast forwarding on the interface (not applicable to a switch).

```
Ruijie(config-if)# ip ref
Ruijie(config-if)# ip vrf forwarding customer_vpn1
Ruijie(config-if)# ip address 10.0.1.1 255.255.255.0
Ruijie(config-if)# no shutdown
Ruijie(config-if)# exit
Ruijie(config)# router ospf 10 vrf customer_vpn1
Ruijie(config-router)# network 10.0.1.0 0.0.0.255 area 0
Ruijie(config-router)# redistribute bgp 65010 subnets
Ruijie(config-router)# exit
Ruijie(config)# router bgp 65010
Ruijie(config-router)# address-family ipv4 vrf customer_vpn1
Ruijie(config-router-af)# redistribute ospf 10 vrf customer_vpn1
Ruijie(config-router-af)# exit
Ruijie(config-router)# exit
```

# On CE3, use the following command.

```
Ruijie(config)# interface gigabitEthernet 0/0
```

# In case of a switch, configure the interface to a RoutedPort interface (not applicable to a router).

```
Ruijie(config-if)# no switchport
```

# In case of a router, enable fast forwarding on the interface (not applicable to a switch).

```
Ruijie(config-if)# ip ref
Ruijie(config-if)# ip address 10.0.1.2 255.255.255.0
Ruijie(config-if)# no shutdown
Ruijie(config)# interface gigabitEthernet 0/1
```

# In case of a switch, configure the interface to a RoutedPort interface (not applicable to a router).

```
Ruijie(config-if)# no switchport
```

# In case of a router, enable fast forwarding on the interface (not applicable to a switch).

```
Ruijie(config-if)# ip ref
Ruijie(config-if)# ip address 61.10.55.1 255.255.255.0
Ruijie(config-if)# no shutdown
Ruijie(config-if)# exit
Ruijie(config)# router ospf 1
Ruijie(config-router)# network 10.0.1.0 0.0.0.255 area 0
Ruijie(config-router)# network 61.10.55.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

Verification

# Display VRF routes and labels on the first carrier PE: PE1 is used as an example. The verification of PE2 is similar.

// In the VRF routing table of PE1, there are only internal routes of the second carrier. There is no VPN route (for example, 64.30.4.0/24).

```
Ruijie# show ip route vrf vpn1
Routing Table: vpn1

Codes:  C - connected, S - static, R - RIP, B - BGP
```

```
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
B    3.3.3.3/32 [200/0] via 172.1.1.2, 00:00:07
C    172.1.1.0/24 is directly connected, gigabitEthernet 1/1
C    172.1.1.1/32 is local host.
B    172.1.2.0/24 [200/0] via 172.1.1.2, 00:00:07
B    172.1.4.0/24 [200/0] via 2.2.2.2, 00:00:30
......
Ruijie# show bgp vpnv4 unicast vrf vpn1 labels
BGP table version is 1, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
            S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network         Next Hop       In Label/Out Label
Route Distinguisher: 65001:20 (Default for VRF vpn1)
*> 3.3.3.3/32      172.1.1.2       2048/1024
*> 172.1.2.0/24    172.1.1.2      2049/1025
*>i6.6.6.6/32      2.2.2.2         2050/2112
......
```

# In the VRF of the second carrier PE and user VPN CE, display the routing table.

// On the PE (using PE3 as an example), use the following command.

```
Ruijie# show ip route vrf customer_vpn1
Routing Table: customer_vpn1

Codes:  C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
......
O    61.10.55.0/24 [200/0] via 10.0.1.2, 00:00:40, gigabitEthernet 1/2
B    64.21.33.0/24 [200/0] via 7.7.7.7, 00:00:31
......
```

// On the user VPN CE (using CE3 as an example), use the following command.

```
Ruijie# show ip route
Codes:  C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
......
C    61.10.55.0/24 is directly connected, gigabitEthernet
C    61.10.55.1/32 is local host.
O    64.21.33.0/24 [200/0] via 10.0.1.1, 00:00:42, gigabitEthernet 0/0
......
```

# Verify that the user VPN networks are interconnected.

//On CE3, use the following command.

```
Ruijie# ping 64.21.33.9
Sending 5, 100-byte ICMP Echoes to 64.21.33.9, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 10/20/40 ms
```

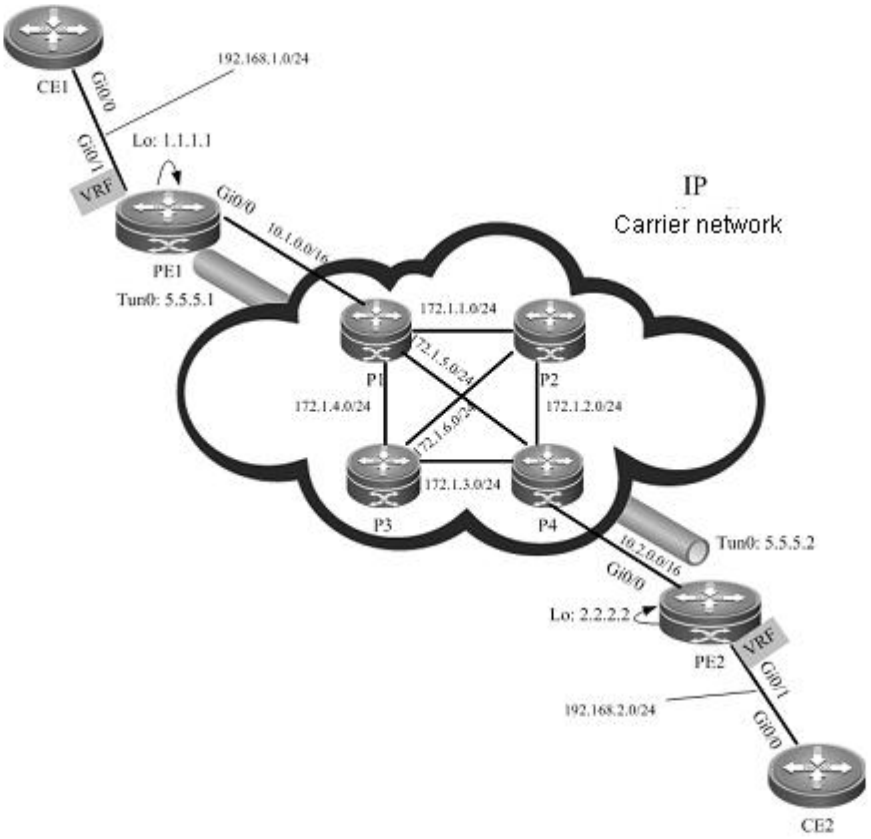
MPLS VPN over GRE

Networking Requirements

In an IP core network, the edge router PE supports the MPLS VPN. Now it is required to use the MPLS VPN over GRE technology to use the IP core network to provide MPLS VPN services for users. The IP core network adopts dual OSPF instances to introduce VPN traffic into the GRE tunnel, while the PE and CE exchange routes via OSPF.

Networking Topology

Figure 42 Network topology



Configuration Tips

- Creating a GRE tunnel
- Configuring an IGP route
- Configuring an MPLS network
- Configuring an MPLS VPN

Configuration Steps

- Configure the P device. P1 is used as an example. The configurations of other devices are similar.

# Configure an interface and IP address.

```
Ruijie(config)# interface gigabitEthernet 0/0
```

# In case of a switch, configure the interface to a RoutedPort interface (not applicable to a router).

```
Ruijie(config-if)# no switchport
```

# In case of a router, enable fast forwarding on the interface (not applicable to a switch).

```
Ruijie(config-if)# ip ref
Ruijie(config-if)# ip address 10.1.0.2 255.255.0.0
Ruijie(config-if)# no shutdown
Ruijie(config-if)# exit
Ruijie(config)# interface gigabitEthernet 0/1
```

# In case of a switch, configure the interface to a RoutedPort interface (not applicable to a router).

```
Ruijie(config-if)# no switchport
```

# In case of a router, enable fast forwarding on the interface (not applicable to a switch).

```
Ruijie(config-if)# ip ref
Ruijie(config-if)# ip address 172.1.1.1 255.255.255.0
Ruijie(config-if)# no shutdown
Ruijie(config-if)# exit
Ruijie(config)# interface gigabitEthernet 1/1
```

# In case of a switch, configure the interface to a RoutedPort interface (not applicable to a router).

```
Ruijie(config-if)# no switchport
```

# In case of a router, enable fast forwarding on the interface (not applicable to a switch).

```
Ruijie(config-if)# ip ref
Ruijie(config-if)# ip address 172.1.5.1 255.255.255.0
Ruijie(config-if)# no shutdown
Ruijie(config-if)# exit
Ruijie(config)# interface gigabitEthernet 0/3
```

# In case of a switch, configure the interface to a RoutedPort interface (not applicable to a router).

```
Ruijie(config-if)# no switchport
```

# In case of a router, enable fast forwarding on the interface (not applicable to a switch).

```
Ruijie(config-if)# ip ref
Ruijie(config-if)# ip address 172.1.4.1 255.255.255.0
Ruijie(config-if)# no shutdown
Ruijie(config-if)# exit
```

# Configure an IGP routing instance.

```
Ruijie(config)# router ospf 1
Ruijie(config-router)# network 172.1.1.0 0.0.0.255 area 0
Ruijie(config-router)# network 172.1.5.0 0.0.0.255 area 0
Ruijie(config-router)# network 172.1.4.0 0.0.0.255 area 0
Ruijie(config-router)# network 10.1.0.0 0.0.255.255 area 0
Ruijie(config-router)# exit
```

- Configure the PE device. PE1 is used as an example. The configurations of other devices are similar.

# Configure a public network interface and IP address.

```
Ruijie(config)# interface Loopback 0
Ruijie(config-if)# ip address 1.1.1.1 255.255.255.255
Ruijie(config-if)# exit
Ruijie(config)# interface gigabitEthernet 0/0
```

# In case of a switch, configure the interface to a RoutedPort interface (not applicable to a router).

```
Ruijie(config-if)# no switchport
```

# In case of a router, enable fast forwarding on the interface (not applicable to a switch).

```
Ruijie(config-if)# ip ref
Ruijie(config-if)# ip address 10.1.0.1 255.255.0.0
Ruijie(config-if)# no shutdown
Ruijie(config-if)# exit
```

# Create a GRE tunnel.

```
Ruijie(config)# interface tunnel 0
Ruijie(config-if)# ip address 5.5.5.1 255.255.255.0
Ruijie(config-if)# tunnel mode gre ip
Ruijie(config-if)# tunnel source 10.1.0.1
Ruijie(config-if)# tunnel destination 10.2.0.1
Ruijie(config-if)# exit
```

# Configure an IGP route.

```
Ruijie(config)# router ospf 1
Ruijie(config-router)# network 10.1.0.0 0.0.255.255 area 0
Ruijie(config-router)# exit
Ruijie(config)# router ospf 2
Ruijie(config-router)# network 1.1.1.1 0.0.0.0 area 0
Ruijie(config-router)# network 5.5.5.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure an MPLS network.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# exit
Ruijie(config)# interface tunnel 0
```

# In case of a router, enable fast forwarding on the interface (not applicable to a switch).

```
Ruijie(config-if)# ip ref
Ruijie(config-if)# mpls ip
Ruijie(config-if)# label-switching
Ruijie(config-if)# exit
```

# Configure an MPLS VPN.

## Configure a VRF.

```
Ruijie(config)# ip vrf vpn1
Ruijie(config-vrf)# rd 100:1
Ruijie(config-vrf)# route-target both 100:1
Ruijie(config-vrf)# exit
Ruijie(config)# interface gigabitEthernet 0/1
```

# In case of a switch, configure the interface to a RoutedPort interface (not applicable to a router).

```
Ruijie(config-if)# no switchport
```

# In case of a router, enable fast forwarding on the interface (not applicable to a switch).

```
Ruijie(config-if)# ip ref
Ruijie(config-if)# ip vrf forwarding vpn1
Ruijie(config-if)# ip address 192.168.1.1 255.255.255.0
Ruijie(config-if)# no shutdown
Ruijie(config-if)# exit
```

## Configure MP-IBGP.

```
Ruijie(config)# router bgp 100
Ruijie(config-router)# neighbor 2.2.2.2 remote-as 100
Ruijie(config-router)# neighbor 2.2.2.2 update-source Loopback 0
Ruijie(config-router)# address-family vpnv4
Ruijie(config-router-af)# neighbor 2.2.2.2 active
Ruijie(config-router-af)# neighbor 2.2.2.2 send-community both
Ruijie(config-router-af)# exit
Ruijie(config-router)# address-family ipv4 vrf vpn1
Ruijie(config-router-af)# redistribute ospf 10 vrf vpn1
Ruijie(config-router-af)# exit
Ruijie(config-router)# exit
```

## Configure route exchange between PEs and CEs.

```
Ruijie(config)# router ospf 10 vrf vpn1
Ruijie(config-router)# network 192.168.1.0 0.0.0.255 area 0
Ruijie(config-router)# redistribute bgp 100 subnets
Ruijie(config-router)# exit
```

- Configure the CE. CE1 is used as an example. The configurations of CE2 are similar.

```
Ruijie(config)# interface gigabitEthernet 0/0
```

# In case of a router, enable fast forwarding on the interface (not applicable to a switch).

```
Ruijie(config-if)# ip ref
Ruijie(config-if)# ip address 192.168.1.2 255.255.255.0
Ruijie(config-if)# no shutdown
Ruijie(config-if)# exit
Ruijie(config)# router ospf 1
Ruijie(config-router)# network 192.168.1.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

Verification

# On the PE, check routing entries. PE1 is used as an example. The next-hop interface of route 2.2.2.2/32 is Tunnel 0.

```
Ruijie# show ip route
Codes:  C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
C    10.1.0.0/16 is directly connected, gigabitEthernet 0/0
C    10.1.0.1/32 is local host.
C    1.0.0.0/8 is subnetted
C      1.1.1.1/32 is local host.
O    2.0.0.0/8 is subnetted
O      2.2.2.2/32 [110/11] via 5.5.5.2, 00:00:40, Tunnel 0
......
```

# Verify the VPN route on PE. PE1 is used as an example.

```
Ruijie# show ip route vrf vpn1
Routing Table: vpn1

Codes:  C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
C    192.168.1.0/24 is directly connected, gigabitEthernet 0/1
C    192.168.1.1/32 is local host.
B    192.168.2.0/24 [200/0] via 2.2.2.2, 00:00:41
......
```

# Check MPLS forwarding entries on the PE. PE1 is used as an example.

```
Ruijie# show mpls forwarding-table
Label Operation Code:
PH--PUSH label
PP--POP label
SW--SWAP label
SP—-SWAP topmost label and push new label
DP—-DROP packet
PC—-POP label and continue lookup( IP or Label )
PI--POP label and do ip lookup forward
PN--POP label and forward to nexthop
PM--POP label and do MAC lookup forward
PV--POP label and output to VC attach interface
IP--IP lookup forward
Local  Outgoing  OP  FEC        Outgoing      Next Hop
label  label                   interface
```

```
--    3         PH 2.2.2.2/32  Tunnel 0      5.5.5.2
--    21        PH 192.168.2.0/24(V) Tunnel 0  Point2point
......
```

# Verify the routing table on the CE.

```
Ruijie# show ip route
Codes:  C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
C    192.168.1.0/24 is directly connected, gigabitEthernet 0/1
C    192.168.1.2/32 is local host.
O    192.168.2.0/24 [112/11] via 192.168.1.1, 00:00:41
......
```

# Verify the intercommunication between CEs. On CE1, use the following command.

```
Ruijie# ping 192.168.2.2
Sending 5, 100-byte ICMP Echoes to 192.168.2.2, timeout is 2 seconds:
  < press Ctrl+C to break >
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 10/20/40 ms
```

OSPF VPN Configuration Examples

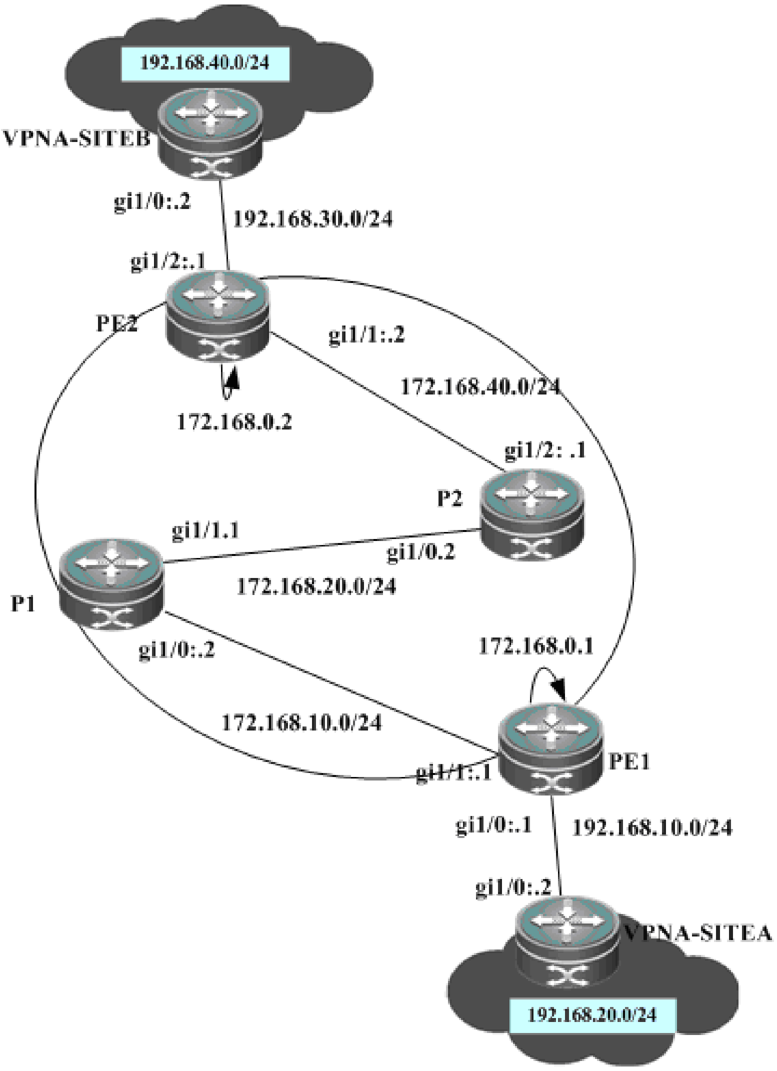
Domain-id Configuration Example

Networking Requirements

Two different sites of a customer exchange VPN routes via an MPLS backbone network. The customer's sites are connected to a PE via OSPF. It is required that the customer's OSPF routes can be restored to the OSPF routes of original sites after being exchanged over the MPLS backbone network.

Networking Topology

Figure 43



To meet the requirements, configure two VRF OSPF instances with the same domain ID on two PEs, as shown below:

Configuration Steps

SITEA:

# Configure the OSPF protocol between the PE and the CE.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 192.168.10.0 255.255.255.0 area 0
```

PE1:

# Configure a loopback interface.

```
Ruijie# configure terminal
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 172.168.0.1 255.255.255.255
```

# Configure a VRF.

Create a VRF named VPNA, set the RD and RT values, and associate the VRF with the corresponding interface.

```
Ruijie# configure terminal
Ruijie(config)# ip vrf VPNA
Ruijie(config-vrf)# rd 1:100
Ruijie(config-vrf)# route-target both 1:100
Ruijie(config-vrf)# end
```

# Associate the CE-connecting interface with the VRF.

```
Ruijie# configure terminal
Ruijie(config)# interface gigabitethernet 1/0
```

# In case of a switch, configure the interface to a RoutedPort interface (not applicable to a router).

```
Ruijie(config-if-GigabitEthernet 1/0)# no switchport
```

# In case of a router, enable fast forwarding on the interface (not applicable to a switch).

```
Ruijie(config-if-GigabitEthernet 1/0)# ip ref
Ruijie(config-if-GigabitEthernet 1/0)# ip vrf forwarding VPNA
Ruijie(config-if-GigabitEthernet 1/0)# ip address 192.168.10.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/0)# end
```

# Configure the BGP protocol to set up an MP-IBGP session with PE2.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 172.168.0.2 remote-as 1
Ruijie(config-router)# neighbor 172.168.0.2 update-source loopback 0
Ruijie(config-router)# address-family vpnv4
Ruijie(config-router-af)# neighbor 172.168.0.2 activate
Ruijie(config-router-af)# end
```

# Exchange routes with the CE via OSPF; set the domain ID of the OSPF instance to 10.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 10 VPNA
Ruijie(config-router)# network 192.168.10.0 255.255.255.0 area 0
Ruijie(config-router)# redistribute bgp subnets
Ruijie(config-router)# domain-id 10
Ruijie(config-router)# exit
Ruijie(config)# router bgp 1
Ruijie(config-router)# address-family ipv4 vrf VPNA
Ruijie(config-router-af)# redistribute ospf 10
Ruijie(config-router-af)# end
```

# Configure MPLS signaling on the backbone network. Enable MPLS on the public network interface.

```
Ruijie# configure terminal
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# exit
Ruijie(config)# interface gigabitethernet 1/1
```

# In case of a switch, configure the interface to a RoutedPort interface (not applicable to a router).

```
Ruijie(config-if-GigabitEthernet 1/1)# no switchport
```

# In case of a router, enable fast forwarding on the interface (not applicable to a switch).

```
Ruijie(config-if-GigabitEthernet 1/1)# ip ref
Ruijie(config-if-GigabitEthernet 1/1)# ip address 172.168.10.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/1)# label-switching
Ruijie(config-if-GigabitEthernet 1/1)# mpls ip
Ruijie(config-if-GigabitEthernet 1/1)# end
```

# Configure a routing protocol on the backbone network.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 1
Ruijie(config-router)# network 172.168.10.0 0.0.0.255 area 0
Ruijie(config-router)# network 172.168.0.1 0.0.0.0 area 0
Ruijie(config-router)# end
```

P1 and P2:The configuration steps are similar to the configuration steps of P in the MPLS backbone network.

SITEB:

# Run OSPF with PE2.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 192.168.30.0 255.255.255.0 area 0
```

PE2:

# Configure a loopback interface.

```
Ruijie# configure terminal
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 172.168.0.2 255.255.255.255
```

# Configure a VRF.

Create a VRF named VPNA, set the RD and RT values, and associate the VRF with the corresponding interface.

```
Ruijie# configure terminal
Ruijie(config)# ip vrf VPNA
Ruijie(config-vrf)# rd 1:100
Ruijie(config-vrf)# route-target both 1:100
Ruijie(config-vrf)# exit
```

# Associate the CE-connecting interface with the VRF.

```
Ruijie# configure terminal
Ruijie(config)# interface gigabitethernet 1/2
```

# In case of a switch, configure the interface to a RoutedPort interface (not applicable to a router).

```
Ruijie(config-if-GigabitEthernet 1/2)# no switchport
```

# In case of a router, enable fast forwarding on the interface (not applicable to a switch).

```
Ruijie(config-if-GigabitEthernet 1/2)# ip ref
Ruijie(config-if-GigabitEthernet 1/2)# ip vrf forwarding VPNA
Ruijie(config-if-GigabitEthernet 1/2)# ip address 192.168.30.1
255.255.255.0
Ruijie(config-if-GigabitEthernet 1/0)# exit
```

# Configure the BGP protocol to set up an MP-IBGP session with PE2.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 172.168.0.1 remote-as 1
Ruijie(config-router)# neighbor 172.168.0.1 update-source loopback 0
Ruijie(config-router)# address-family vpnv4
Ruijie(config-router-af)# neighbor 172.168.0.1 activate
Ruijie(config-router-af)# end
```

# Exchange VPN routes with the CE via OSPF; set the domain ID to 10.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 10 VPNA
Ruijie(config-router)# network 192.168.30.0 255.255.255.0 area 0
Ruijie(config-router)# redistribute bgp subnets
Ruijie(config-router)# domain-id 10
Ruijie(config-router)# exit
Ruijie(config)# router bgp 1
Ruijie(config-router)# address-family ipv4 vrf VPNA
Ruijie(config-router-af)# redistribute ospf 10
Ruijie(config-router-af)# exit
```

# Configure MPLS signaling on the backbone network. Enable MPLS on the public network interface.

```
Ruijie# configure terminal
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# exit
Ruijie(config)# interface gigabitethernet 1/1
```

# In case of a switch, configure the interface to a RoutedPort interface (not applicable to a router).

```
Ruijie(config-if-GigabitEthernet 1/1)# no switchport
```

# In case of a router, enable fast forwarding on the interface (not applicable to a switch).

```
Ruijie(config-if-GigabitEthernet 1/1)# ip ref
Ruijie(config-if-GigabitEthernet 1/1)# ip address 172.168.40.2 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/1)# label-switching
Ruijie(config-if-GigabitEthernet 1/1)# mpls ip
Ruijie(config-if-GigabitEthernet 1/1)# end
```

# Configure a routing protocol on the backbone network.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 1
Ruijie(config-router)# network 172.168.40.0 0.0.0.255 area 0
Ruijie(config-router)# network 172.168.0.2 0.0.0.0 area 0
Ruijie(config-router)# end
```

Verification

- VPNA-SITEB:

```
Ruijie# show ip route
Codes:  C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
O IA    192.168.10.0/24 [110/2] via 192.168.30.1, 00:00:36, GigabitEthernet 1/0
O IA    192.168.20.0/24 [110/2] via 192.168.30.1, 00:00:36, GigabitEthernet 1/0
C      192.168.30.0/24 is directly connected, GigabitEthernet 1/0
O      192.168.40.0/24 [110/101] via 192.168.24.2, 00:56:23, GigabitEthernet 1/1
```

- PE2:

```
Ruijie# show ip route vrf VPNA
Routing Table: VPNA

Codes:  C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
B      192.168.10.0/24 [110/2] via 172.168.0.1, 00:00:36
B      192.168.20.0/24 [110/2] via 172.168.0.1, 00:00:36
C      192.168.30.0/24 is directly connected, GigabitEthernet 1/2
O      192.168.40.0/24 [110/101] via 192.168.30.2, 00:56:23, GigabitEthernet 1/2
```

- PE1:

```
Ruijie# show ip route vrf VPNA
```

```
Routing Table: VPNA

Codes:  C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
C      192.168.10.0/24 is directly connected, GigabitEthernet 1/0
O      192.168.20.0/24 [110/101] via 192.168.10.2, 00:56:23, GigabitEthernet 1/0
B      192.168.30.0/24 [110/2] via 172.168.0.2, 00:00:36
B      192.168.40.0/24 [110/2] via 172.168.0.2, 00:00:36
```

- VPNA-SITEA:

```
Ruijie# show ip route
Codes:  C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
C      192.168.10.0/24 is directly connected, GigabitEthernet 1/0
O      192.168.20.0/24 [110/101] via 192.168.23.2, 00:56:23, GigabitEthernet 1/1
O IA    192.168.30.0/24 [110/2] via 192.168.10.1, 00:00:36, GigabitEthernet 1/0
O IA    192.168.40.0/24 [110/2] via 192.168.10.1, 00:00:36, GigabitEthernet 1/0
```

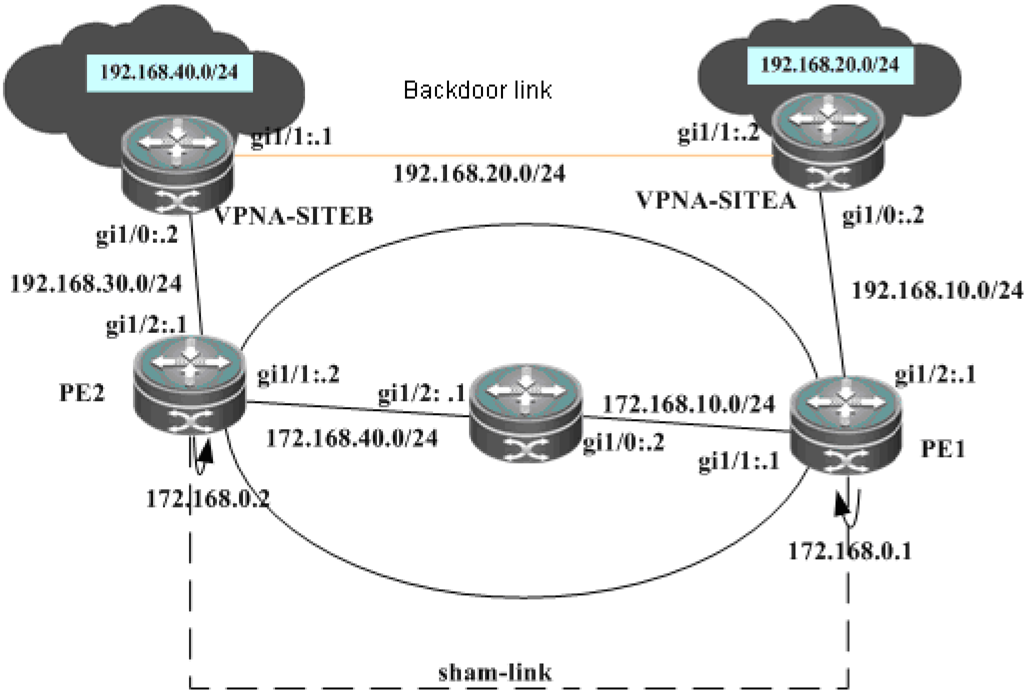
Sham Link Configuration Example

Networking Requirements

Two different sites of a customer exchange VPN routes via an MPLS backbone network. At the same time, a "backdoor link" is also established between these two sites to ensure that information can still be exchanged between both sites through this backup link when the MPLS backbone network fails.

Networking Topology

Figure 44



Configuration Steps

SITEA:

# Run the OSPF protocol with PE1 and SITEB. The OSPF protocol runs over the backdoor link with SITEB.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 192.168.10.0 255.255.255.0 area 0
Ruijie(config-router)# network 192.168.20.0 255.255.255.0 area 0
```

# Configure the OSPF cost on an interface.

```
Ruijie# configure terminal
Ruijie(config)# interface gigabitethernet 1/0
```

# In case of a switch, configure the interface to a RoutedPort interface (not applicable to a router).

```
Ruijie(config-if-GigabitEthernet 1/0)# no switchport
```

# In case of a router, enable fast forwarding on the interface (not applicable to a switch).

```
Ruijie(config-if-GigabitEthernet 1/0)# ip ref
Ruijie(config-if-GigabitEthernet 1/0)# ip address 192.168.10.2 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/0)# ip ospf cost 1
Ruijie(config)# interface gigabitethernet 1/1
```

# In case of a switch, configure the interface to a RoutedPort interface (not applicable to a router).

```
Ruijie(config-if-GigabitEthernet 1/1)# no switchport
```

# In case of a router, enable fast forwarding on the interface (not applicable to a switch).

```
Ruijie(config-if-GigabitEthernet 1/1)# ip ref
Ruijie(config-if-GigabitEthernet 1/1)# ip address 192.168.20.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/1)# ip ospf cost 200
```

PE1:

# Configure a loopback interface.

```
Ruijie# configure terminal
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 172.168.0.1 255.255.255.255
```

# Configure a VRF.

Create a VRF named VPNA, set the RD and RT values, and associate the VRF with the corresponding interface.

```
Ruijie# configure terminal
Ruijie(config)# ip vrf VPNA
Ruijie(config-vrf)# rd 1:100
Ruijie(config-vrf)# route-target both 1:100
Ruijie(config-vrf)# end
```

# Associate the CE-connecting interface with the VRF.

```
Ruijie# configure terminal
Ruijie(config)# interface gigabitethernet 1/2
```

# In case of a switch, configure the interface to a RoutedPort interface (not applicable to a router).

```
Ruijie(config-if-GigabitEthernet 1/2)# no switchport
```

# In case of a router, enable fast forwarding on the interface (not applicable to a switch).

```
Ruijie(config-if-GigabitEthernet 1/2)# ip ref
Ruijie(config-if-GigabitEthernet 1/2)# ip vrf forwarding VPNA
Ruijie(config-if-GigabitEthernet 1/2)# ip address 192.168.10.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/2)# end
```

# Configure the VRF loopback interface to establish a sham link.

```
Ruijie# configure terminal
Ruijie(config)# interface loopback 10
Ruijie(config-if-Loopback 10)# ip vrf forwarding VPNA
Ruijie(config-if-Loopback 10)# ip address 192.168.0.1 255.255.255.255
```

# Configure the BGP protocol to set up an MP-IBGP session with PE2.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 172.168.0.2 remote-as 1
Ruijie(config-router)# neighbor 172.168.0.2 update-source loopback 0
Ruijie(config-router)# address-family vpnv4
Ruijie(config-router-af)# neighbor 172.168.0.2 activate
Ruijie(config-router-af)# end
```

# Exchange routes with the CE via OSPF, and establish a sham link with the OSPF instance on PE2.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 10 VPNA
Ruijie(config-router)# network 192.168.10.0 255.255.255.0 area 0
Ruijie(config-router)# redistribute bgp subnets
Ruijie(config-router)# area 0 sham-link 192.168.0.1 192.168.0.2
Ruijie(config-router)# exit
Ruijie(config)# router bgp 1
Ruijie(config-router)# address-family ipv4 vrf VPNA
Ruijie(config-router-af)# redistribute ospf 10
Ruijie(config-router-af)# redistribute connected
Ruijie(config-router-af)# end
```

# Configure MPLS signaling on the backbone network. Enable MPLS on the public network interface.

```
Ruijie# configure terminal
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# exit
Ruijie(config)# interface gigabitethernet 1/1
```

# In case of a switch, configure the interface to a RoutedPort interface (not applicable to a router).

```
Ruijie(config-if-GigabitEthernet 1/1)# no switchport
```

# In case of a router, enable fast forwarding on the interface (not applicable to a switch).

```
Ruijie(config-if-GigabitEthernet 1/1)# ip ref
Ruijie(config-if-GigabitEthernet 1/1)# ip address 172.168.10.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/1)# label-switching
Ruijie(config-if-GigabitEthernet 1/1)# mpls ip
Ruijie(config-if-GigabitEthernet 1/1)# end
```

# Configure a routing protocol on the backbone network.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 1
Ruijie(config-router)# network 172.168.10.0 0.0.0.255 area 0
Ruijie(config-router)# network 172.168.0.1 0.0.0.0 area 0
Ruijie(config-router)# end
```

P1:

The configuration steps are similar to the configuration steps of P in the MPLS backbone network.

SITEB:

# Run the OSPF protocol with PE2 and SITEA. The OSPF protocol runs over the backup link with SITEA.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 192.168.30.0 255.255.255.0 area 0
Ruijie(config-router)# network 192.168.20.0 255.255.255.0 area 0
```

# Configure the OSPF cost on an interface.

```
Ruijie# configure terminal
Ruijie(config)# interface gigabitethernet 1/0
```

# In case of a switch, configure the interface to a RoutedPort interface (not applicable to a router).

```
Ruijie(config-if-GigabitEthernet 1/0)# no switchport
```

# In case of a router, enable fast forwarding on the interface (not applicable to a switch).

```
Ruijie(config-if-GigabitEthernet 1/0)# ip ref
Ruijie(config-if-GigabitEthernet 1/0)# ip address 192.168.30.2 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/0)# ip ospf cost 1
Ruijie(config)# interface gigabitethernet 1/1
```

# In case of a switch, configure the interface to a RoutedPort interface (not applicable to a router).

```
Ruijie(config-if-GigabitEthernet 1/1)# no switchport
```

# In case of a router, enable fast forwarding on the interface (not applicable to a switch).

```
Ruijie(config-if-GigabitEthernet 1/1)# ip ref
Ruijie(config-if-GigabitEthernet 1/1)# ip address 192.168.20.2 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/1)# ip ospf cost 200
```

PE2:

# Configure a loopback interface.

```
Ruijie# configure terminal
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 172.168.0.2 255.255.255.255
```

# Configure a VRF.

Create a VRF named VPNA, set the RD and RT values, and associate the VRF with the corresponding interface.

```
Ruijie# configure terminal
Ruijie(config)# ip vrf VPNA
Ruijie(config-vrf)# rd 1:100
Ruijie(config-vrf)# route-target both 1:100
Ruijie(config-vrf)# exit
```

# Associate the CE-connecting interface with the VRF.

```
Ruijie# configure terminal
Ruijie(config)# interface gigabitethernet 1/2
```

# In case of a switch, configure the interface to a RoutedPort interface (not applicable to a router).

```
Ruijie(config-if-GigabitEthernet 1/2)# no switchport
```

# In case of a router, enable fast forwarding on the interface (not applicable to a switch).

```
Ruijie(config-if-GigabitEthernet 1/2)# ip ref
Ruijie(config-if-GigabitEthernet 1/2)# ip vrf forwarding VPNA
Ruijie(config-if-GigabitEthernet 1/2)# ip address 192.168.30.1
255.255.255.0
Ruijie(config-if-GigabitEthernet 1/2)# exit
```

# Configure the VRF loopback interface to establish a sham link.

```
Ruijie# configure terminal
Ruijie(config)# interface loopback 10
Ruijie(config-if-Loopback 10)# ip vrf forwarding VPNA
Ruijie(config-if-Loopback 10)# ip address 192.168.0.2 255.255.255.255
```

# Configure the BGP protocol to set up an MP-IBGP session with PE2 and PE3.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 172.168.0.1 remote-as 1
Ruijie(config-router)# neighbor 172.168.0.1 update-source loopback 0
Ruijie(config-router)# address-family vpnv4
Ruijie(config-router-af)# neighbor 172.168.0.1 activate
Ruijie(config-router-af)# end
```

# Exchange VPN routes with the CE via OSPF, and establish a sham link with PE1.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 10 VPNA
Ruijie(config-router)# network 192.168.30.0 255.255.255.0 area 0
Ruijie(config-router)# redistribute bgp subnets
Ruijie(config-router)# area 0 sham-link  192.168.0.2 192.168.0.1
Ruijie(config-router)# exit
Ruijie(config)# router bgp 1
Ruijie(config-router)# address-family ipv4 vrf VPNA
Ruijie(config-router-af)# redistribute ospf 10
uijie(config-router-af)# redistribute connected
Ruijie(config-router-af)# exit
```

# Configure MPLS signaling on the backbone network. Enable MPLS on the public network interface.

```
Ruijie# configure terminal
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# exit
Ruijie(config)# interface gigabitethernet 1/1
```

# In case of a switch, configure the interface to a RoutedPort interface (not applicable to a router).

```
Ruijie(config-if-GigabitEthernet 1/1)# no switchport
```

# In case of a router, enable fast forwarding on the interface (not applicable to a switch).

```
Ruijie(config-if-GigabitEthernet 1/1)# ip ref
Ruijie(config-if-GigabitEthernet 1/1)# ip address 172.168.40.2 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/1)# label-switching
Ruijie(config-if-GigabitEthernet 1/1)# mpls ip
Ruijie(config-if-GigabitEthernet 1/1)# end
```

# Configure a routing protocol on the backbone network.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 1
Ruijie(config-router)# network 172.168.40.0 0.0.0.255 area 0
Ruijie(config-router)# network 172.168.0.2 0.0.0.0 area 0
Ruijie(config-router)# end
```

Verification

- PE1

```
Ruijie# show ip ospf 10 sham-links
Sham Link OSPF_SL0 to address 192.168.0.2 is up
Area 0 source address 192.168.0.1
  Run as demand circuit
  DoNotAge LSA allowed. Cost of using 1 State POINT_TO_POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40,
    Hello due in 00:00:06
    Adjacency State FULL (Hello suppressed)
    Index 2/2, retransmission queue length 0, number of retransmission 0
    First 0x0(0)/0x0(0) Next 0x0(0)/0x0(0)
    Last retransmission scan length is 0, maximum is 0
Last retransmission scan time is 0 msec, maximum is 0 msec
Ruijie# show ip ospf 10 neighbor
Neighbor ID     Pri   State          Dead Time   Address        Interface
192.168.0.2       0   FULL/  -          -       192.168.0.2    OSPF_SL0
Ruijie# show ip route vrf VPNA
Codes:  C - connected, S - static, R - RIP, B - BGP
        O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default
C       192.168.10.0/24 is directly connected, Gi1/2
O       192.168.20.0/24 [110/101] via 192.168.1.2, 00:56:23, Gi1/2
O       192.168.40.0/24 [110/2] via 172.168.0.2, 00:00:36
```

- PE2

```
Ruijie# show ip ospf 10 sham-links
Sham Link SLINK0 to address 192.168.0.1 is up
  Area 0.0.0.0 source address 192.168.0.2, Cost: 1
  Output interface is GigabitEthernet 1/1
  Nexthop address 172.16.10.1
  Transmit Delay is 1 sec, State Point-To-Point,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
   Hello due in 00:00:01
   Adjacency state Full

Ruijie# show ip ospf 10 neighbor

OSPF process 10, 1 Neighbors, 1 is Full:
Neighbor ID    Pri   State             BFD State  Dead Time   Address       Interface
192.168.0.1       1   Full/ -            -        00:00:34   192.168.0.1       SLINK0

Ruijie# show ip route vrf VPNA
Routing Table: VPNA

Codes:  C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default

Gateway of last resort is no set
O      192.168.10.0/24 [110/2] via 172.168.0.1, 00:00:36
O      192.168.20.0/24 [110/2] via 172.168.0.1, 00:00:36
C      192.168.30.0/24 is directly connected, GigabitEthernet 1/2
O      192.168.40.0/24 [110/101] via 192.168.30.2, 00:56:23, GigabitEthernet 1/2
```

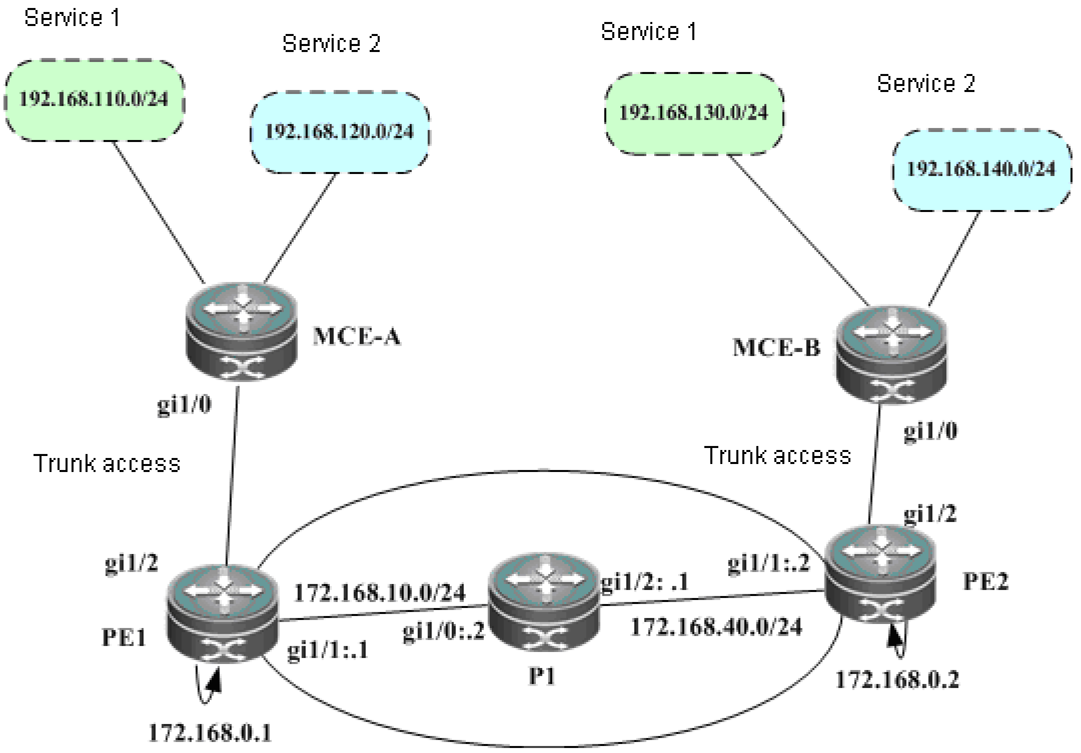
Configuring Multiple OSPF Instances on an MCE

Networking Requirements

A customer site involves multiple services. The same services communicate with each other over an MPLS backbone network and different services are isolated from each other.

Networking Topology

Figure 45



Configuration Steps

MCE-A:

# Configure a trunk link between a PE and a CE.

```
Ruijie# configure terminal
Ruijie(config)# interface gigabitethernet 1/0
```

# **VLAN** is a configuration command used on switch products, and is not applicable to routers (except for the switching card interface). Routers can use the routing interface to connect to the PE.

```
Ruijie(config-if-GigabitEthernet 1/0)# switchport mode trunk
Ruijie(config-if-GigabitEthernet 1/0)# end
```

# Configure two VRFs to represent different services and bind respective interfaces.

```
Ruijie# configure terminal
Ruijie(config)# ip vrf VPN1
Ruijie(config-vrf)# exit
Ruijie(config)# VLAN 10
Ruijie(config)# interface vlan 10
```

# **VLAN** is a configuration command used on switch products, and is not applicable to routers (except for the switching card interface). Routers can use the sub-interface to bind the VRF.

```
Ruijie(config-if-vlan 10)# ip vrf forwarding VPN1
Ruijie(config-if-vlan 10)# ip address 192.168.10.2 255.255.255.0
Ruijie(config)# ip vrf VPN2
Ruijie(config-vrf)# exit
Ruijie(config)# VLAN 20
Ruijie(config)# interface vlan 20
```

# **VLAN** is a configuration command used on switch products, and is not applicable to routers (except for the switching card interface). Routers can use the sub-interface to bind the VRF.

```
Ruijie(config-if-vlan 20)# ip vrf forwarding VPN2
Ruijie(config-if-vlan 20)# ip address 192.168.20.2 255.255.255.0
```

# Run the OSPF protocol with the PE for two VRFs.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 10 VPN1
Ruijie(config-router)# network 192.168.10.0 255.255.255.0 area 0
Ruijie(config-router)# capability vrf-lite
```

```
Ruijie(config)# router ospf 10 VPN2
Ruijie(config-router)# network 192.168.20.0 255.255.255.0 area 0
Ruijie(config-router)# capability vrf-lite
```

PE1:

# Configure a loopback interface.

```
Ruijie# configure terminal
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 172.168.0.1 255.255.255.255
```

# Configure the trunk link between the PE and the CE.

```
Ruijie(config)# interface gigabitethernet 1/2
Ruijie(config-if-GigabitEthernet 1/2)# switchport mode trunk
Ruijie(config-if-GigabitEthernet 1/2)# end
```

# Configure VRFs.

Create two VRFs named VPN1 and VPN2 to represent different services, and associate the VRFs with the corresponding interfaces.

```
Ruijie# configure terminal
Ruijie(config)# ip vrf VPN1
Ruijie(config-vrf)# rd 1:100
Ruijie(config-vrf)# route-target both 1:100
Ruijie(config-vrf)# end
Ruijie# configure terminal
Ruijie(config)# ip vrf VPN2
Ruijie(config-vrf)# rd 1:200
Ruijie(config-vrf)# route-target both 1:200
Ruijie(config-vrf)# end
```

# Associate the CE-connecting interface with the VRF.

```
Ruijie(config)# VLAN 10
Ruijie(config)# interface vlan 10
```

# **VLAN** is a configuration command used on switch products, and is not applicable to routers (except for the switching card interface). Routers can use the sub-interface to bind the VRF.

```
Ruijie(config-if-vlan 10)# ip vrf forwarding VPN1
Ruijie(config-if-vlan 10)# ip address 192.168.10.1 255.255.255.0

Ruijie(config)# VLAN 20
Ruijie(config)# interface vlan 20
```

# **VLAN** is a configuration command used on switch products, and is not applicable to routers (except for the switching card interface). Routers can use the sub-interface to bind the VRF.

```
Ruijie(config-if-vlan 20)# ip vrf forwarding VPN1
Ruijie(config-if-vlan 20)# ip address 192.168.20.1 255.255.255.0
```

# Configure the BGP protocol to set up an MP-IBGP session with PE2.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 172.168.0.2 remote-as 1
Ruijie(config-router)# neighbor 172.168.0.2 update-source loopback 0
Ruijie(config-router)# address-family vpnv4
Ruijie(config-router-af)# neighbor 172.168.0.2 activate
Ruijie(config-router-af)# end
```

# Exchange routes with the CE via OSPF.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 10 VPN1
Ruijie(config-router)# network 192.168.10.0 255.255.255.0 area 0
Ruijie(config-router)# exit
Ruijie(config)# router bgp 1
Ruijie(config-router)# address-family ipv4 vrf VPNA
Ruijie(config-router-af)# redistribute ospf 10
Ruijie(config-router-af)# redistribute connected
Ruijie(config-router-af)# end

Ruijie# configure terminal
Ruijie(config)# router ospf 20 VPN2
Ruijie(config-router)# network 192.168.20.0 255.255.255.0 area 0
Ruijie(config-router)# redistribute bgp subnets
Ruijie(config-router)# exit
Ruijie(config)# router bgp 1
Ruijie(config-router)# address-family ipv4 vrf VPN2
Ruijie(config-router-af)# redistribute ospf 20
Ruijie(config-router-af)# redistribute connected
Ruijie(config-router-af)# end
```

# Configure MPLS signaling on the backbone network. Enable MPLS on the public network interface.

```
Ruijie# configure terminal
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# exit
Ruijie(config)# interface gigabitethernet 1/1
```

# In case of a switch, configure the interface to a RoutedPort interface (not applicable to a router).

```
Ruijie(config-if-GigabitEthernet 1/1)# no switchport
```

# In case of a router, enable fast forwarding on the interface (not applicable to a switch).

```
Ruijie(config-if-GigabitEthernet 1/1)# ip ref
Ruijie(config-if-GigabitEthernet 1/1)# ip address 172.168.10.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/1)# label-switching
Ruijie(config-if-GigabitEthernet 1/1)# mpls ip
Ruijie(config-if-GigabitEthernet 1/1)# end
```

# Configure a routing protocol on the backbone network.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 1
Ruijie(config-router)# network 172.168.10.0 0.0.0.255 area 0
Ruijie(config-router)# network 172.168.0.1 0.0.0.0 area 0
Ruijie(config-router)# end
```

P1:

The configuration steps are similar to the configuration steps of P in the MPLS backbone network.

SITEB:

# Configure the trunk link between the PE and the CE.

```
Ruijie# configure terminal
Ruijie(config)# interface gigabitethernet 1/0
```

# **VLAN** is a configuration command used on switch products, and is not applicable to routers (except for the switching card interface). Routers can use the routing interface to connect to the PE.

```
Ruijie(config-if-GigabitEthernet 1/0)# switchport mode trunk
Ruijie(config-if-GigabitEthernet 1/0)# end
```

# Configure two VRFs to represent different services and bind respective interfaces.

```
Ruijie# configure terminal
Ruijie(config)# ip vrf VPN1
Ruijie(config-vrf)# exit
Ruijie(config)# VLAN 10
Ruijie(config)# interface vlan 30
```

# **VLAN** is a configuration command used on switch products, and is not applicable to routers (except for the switching card interface). Routers can use the sub-interface to bind the VRF.

```
Ruijie(config-if-vlan 10)# ip vrf forwarding VPN1
Ruijie(config-if-vlan 10)# ip address 192.168.30.2 255.255.255.0

Ruijie(config)# ip vrf VPN2
Ruijie(config-vrf)# exit
Ruijie(config)# VLAN 40
Ruijie(config)# interface vlan 40
```

# **VLAN** is a configuration command used on switch products, and is not applicable to routers (except for the switching card interface). Routers can use the sub-interface to bind the VRF.

```
Ruijie(config-if-vlan 20)# ip vrf forwarding VPN2
Ruijie(config-if-vlan 20)# ip address 192.168.40.2 255.255.255.0
```

# Run the OSPF protocol with the PE for two VRFs.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 10 VPN1
Ruijie(config-router)# network 192.168.30.0 255.255.255.0 area 0
Ruijie(config-router)# capability vrf-lite

Ruijie(config)# router ospf 10 VPN2
Ruijie(config-router)# network 192.168.40.0 255.255.255.0 area 0
Ruijie(config-router)# capability vrf-lite
```

PE2:

# Configure a loopback interface.

```
Ruijie# configure terminal
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 172.168.0.2 255.255.255.255
```

# Configure the trunk link between the PE and the CE.

```
Ruijie(config)# interface gigabitethernet 1/2
Ruijie(config-if-GigabitEthernet 1/2)# switchport mode trunk
Ruijie(config-if-GigabitEthernet 1/2)# end
```

# Configure VRFs.

Create two VRFs named VPN1 and VPN2 to represent different services, and associate the VRFs with the corresponding interfaces.

```
Ruijie# configure terminal
Ruijie(config)# ip vrf VPN1
Ruijie(config-vrf)# rd 1:100
Ruijie(config-vrf)# route-target both 1:100
Ruijie(config-vrf)# end
```

```
Ruijie# configure terminal
Ruijie(config)# ip vrf VPN2
Ruijie(config-vrf)# rd 1:200
Ruijie(config-vrf)# route-target both 1:200
Ruijie(config-vrf)# end
```

# Associate the CE-connecting interface with the VRF.

```
Ruijie(config)# VLAN 30
Ruijie(config)# interface vlan 30
```

# **VLAN** is a configuration command used on switch products, and is not applicable to routers (except for the switching card interface). Routers can use the sub-interface to bind the VRF.

```
Ruijie(config-if-vlan 10)# ip vrf forwarding VPN1
Ruijie(config-if-vlan 10)# ip address 192.168.30.1 255.255.255.0

Ruijie(config)# VLAN 40
Ruijie(config)# interface vlan 40
```

# **VLAN** is a configuration command used on switch products, and is not applicable to routers. Routers can use the sub-interface to bind the VRF.

```
Ruijie(config-if-vlan 20)# ip vrf forwarding VPN2
Ruijie(config-if-vlan 20)# ip address 192.168.40.1 255.255.255.0
```

# Configure the BGP protocol to set up an MP-IBGP session with PE2.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 172.168.0.1 remote-as 1
Ruijie(config-router)# neighbor 172.168.0.1 update-source loopback 0
Ruijie(config-router)# address-family vpnv4
Ruijie(config-router-af)# neighbor 172.168.0.1 activate
Ruijie(config-router-af)# end
```

# Exchange routes with the CE via OSPF.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 10 VPN1
Ruijie(config-router)# network 192.168.30.0 255.255.255.0 area 0
Ruijie(config-router)# redistribute bgp subnets
Ruijie(config-router)# exit
Ruijie(config)# router bgp 1
Ruijie(config-router)# address-family ipv4 vrf VPN1
Ruijie(config-router-af)# redistribute ospf 10
Ruijie(config-router-af)# redistribute connected
Ruijie(config-router-af)# end

Ruijie# configure terminal
Ruijie(config)# router ospf 20 VPN2
Ruijie(config-router)# network 192.168.40.0 255.255.255.0 area 0
Ruijie(config-router)# redistribute bgp subnets
Ruijie(config-router)# exit
Ruijie(config)# router bgp 1
Ruijie(config-router)# address-family ipv4 vrf VPN2
Ruijie(config-router-af)# redistribute ospf 20
Ruijie(config-router-af)# redistribute connected
Ruijie(config-router-af)# end
```

# Configure MPLS signaling on the backbone network. Enable MPLS on the public network interface.

```
Ruijie# configure terminal
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# exit
Ruijie(config)# interface gigabitethernet 1/1
```

# In case of a switch, configure the interface to a RoutedPort interface (not applicable to a router).

```
Ruijie(config-if-GigabitEthernet 1/1)# no switchport
```

# In case of a router, enable fast forwarding on the interface (not applicable to a switch).

```
Ruijie(config-if-GigabitEthernet 1/1)# ip ref
Ruijie(config-if-GigabitEthernet 1/1)# ip address 172.168.40.2 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/1)# label-switching
Ruijie(config-if-GigabitEthernet 1/1)# mpls ip
Ruijie(config-if-GigabitEthernet 1/1)# end
```

# Configure a routing protocol on the backbone network.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 1
Ruijie(config-router)# network 172.168.40.0 0.0.0.255 area 0
Ruijie(config-router)# network 172.168.0.2 0.0.0.0 area 0
Ruijie(config-router)# end
```

Verification

- MCEA

```
Ruijie# show ip route vrf VPN1
Codes:  C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default
C      192.168.10.0/24 is directly connected, VLAN 10
O      192.168.110.0/24 [110/101] via 192.168.21.2, 00:56:23, Gi1/1
O E2    192.168.130.0/24 [110/2] via 192.168.10.1, 00:00:36, VLAN 10
```

```
Ruijie# show ip route vrf VPN2
Codes:  C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default
C      192.168.20.0/24 is directly connected, VLAN 20
O      192.168.120.0/24 [110/101] via 192.168.22.2, 00:56:23, Gi1/2
O E2   192.168.140.0/24 [110/2] via 192.168.20.1, 00:00:36, VLAN 20
```

- MCEB

```
Ruijie# show ip route vrf VPN1
Codes:  C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default
C      192.168.30.0/24 is directly connected, VLAN 30
O      192.168.130.0/24 [110/101] via 192.168.23.2, 00:56:23, Gi1/1
O E2   192.168.110.0/24 [110/2] via 192.168.30.1, 00:00:36, VLAN 30

Ruijie# show ip route vrf VPN2
Codes:  C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default
C      192.168.40.0/24 is directly connected, VLAN 40
O      192.168.140.0/24 [110/101] via 192.168.24.2, 00:56:23, Gi1/2
O E2   192.168.140.0/24 [110/2] via 192.168.40.1, 00:00:36, VLAN 40
```

Configuring L2VPN

☑ For the router with the L2VPN service enabled, the L2VPN service can be properly forwarded if the fast forwarding function is enabled both on the interfaces connected to the user and the public network.

Understanding L2VPN

Similar to an MPLS/BGP L3VPN, an L2VPN also uses the existing public network to extend the private network of a user. For an MPLS L2VPN, Layer 2 user data (such as ATM cells, FR frames, and Ethernet frames) is transparently transmitted on an MPLS network. As far as the user is concerned, the MPLS network is a Layer 2 switching network where different sites set up Layer 2 connections on the MPLS network.

Compared with L3VPN, L2VPN has the following advantages:

- Supports various link layer protocols. On an MPLS network, you can provide Layer 2 VPN services based on various protocols including ATM, FR, VLAN, Ethernet, PPP, and HDLC. L2VPN also supports multiple network layer protocols, including IP, IPv6, IPX, and SNA.
- A provider edge (PE) does not store the information about user VPNs, reducing the overhead on the PE. This largely lessens the burden of PEs and the entire service provider (SP) network and allows the carriers to support more VPNs and access more users.
- Allows a user to control the advertising of VPN routes and frees the PE from managing VPN routes. This guarantees reliability and confidentiality of user routes and lessens the management burden of carriers.

At present, an MPLS L2VPN has the following two service models:

- Virtual Private Wire Service (VPWS)

You can set up a Pseudo Wire (PW), also called Virtual Circuit (VC), on an IP/MPLS network to simulate Layer 2 point-to-point services, including Ethernet, PPP, HDLC, AAL5 frames, ATM cells, FR, and SONET/SDH. For a user, the VPWS resembles a physical line provided for the user on the carrier's network.

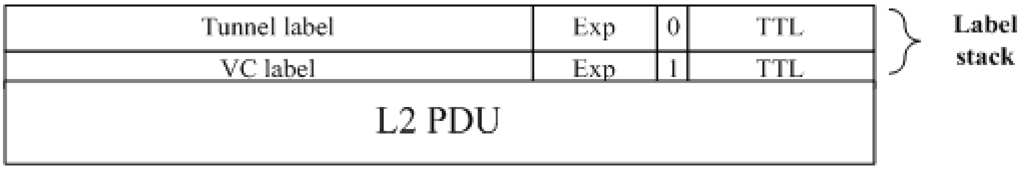
- Virtual Private LAN Service (VPLS)

Simulate LAN services on an IP/MPLS network to implement Ethernet connections on the WAN. For a user, its Layer 2 devices are connected to each other across the IP/MPLS core network and the core network is like a virtual switch.

Frame Format of an MPLS L2VPN Packet

As shown in the following figure, an MPLS L2VPN packet is generally encapsulated in two layers of labels. The outer layer is a public label that is responsible for forwarding the packet on the public network. The inner layer is the VC label that is used to identify a VC instance on a PE.

Figure 46

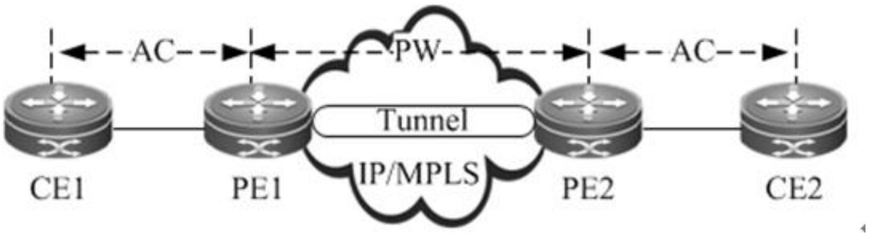


Tunnel label	Exp	0	TTL
VC label	Exp	1	TTL
L2 PDU			

Basic Concepts

The following figure shows the components of an L2VPN.

Figure 47



CE

A custom edge (CE) is a user device directly connected to the SP.

PE

A PE, an edge device on the SP network, is connected to CEs and is responsible for the access of VPN services. It forwards packets from a private network to a public tunnel and from the public tunnel to the private network.
On a hierarchical VPLS network, PEs are classified into User Facing Provider Edge devices (U-PEs) and Network Facing Provider Edge devices (N-PEs).

- U-PE

A U-PE is a PE next to the user side as the hierarchical VPLS network. It is an aggregation device for users to access a VPN.

- N-PE

An N-PE is a core PE device on the hierarchical VPLS network. It is located at the edge of core VPLS domains to provide transparent transmission of VPLS services between core networks.

AC

On an L2VPN of any type (VPWS/VPLS), CEs must be connected to PEs through physical lines or virtual lines. These physical or virtual lines are referred to as the ACs. For example, an AC can be an Ethernet cable, a VLAN, or an MPLS LSP. All user packets on the AC are generally forwarded to the peer CE without any changes.

PW

A PW is responsible for setting up and maintaining the signaling protocol between PEs. The AC transmits frames from a CE to a PE and the PW sends user frames from one PE to another PE.

On a hierarchical VPLS network, PWs are classified into Hub PWs and Spoke PWs.

- Hub PW

A Hub PW indicates a PW set up between N-PEs.

- Spoke PW

A Spoke PW indicates a PW set up between a U-PE and an N-PE, or a PW with which a user accesses the PE on a basic VPLS network.

Forwarder

On an L2VPN, every frame received by a PE from the AC must be forwarded to the corresponding PW. Similarly, the frames received by the PE from the PW are sent to the corresponding AC. This process of making forwarding decisions is called the forwarder.

In VPWS, the forwarder performs the one-to-one mapping between ACs and PWs.

In VPLS, the forwarder is also called the Virtual Switch Instance (VSI) or Virtual Forwarding Instance (VFI), which is the VPLS forwarding table. Through the VFI, you can map the ACs of actual VPLS users to PWs.

Tunnel

The traffic of PWs between PEs is transmitted over the tunnel. One tunnel can carry multiple PWs. The tunnels mentioned in this chapter refer to MPLS LSPs.

Encapsulation

The L2VPN payload is transmitted over PWs. The PWE3 defines the encapsulation formats and transmission technologies of various packets transmitted over PWs. The PW supports two encapsulation modes: raw and tag. In raw mode, the service distinguishers are removed from the PDUs transmitted on PWs. In tag mode, the service distinguishers are included in the PDUs transmitted on PWs. For Ethernet emulated services, the service distinguisher is generally a VLAN tag. Encapsulation and transmission methods of PPP and HDLC packets are defined in the RFC4618.

PW Signaling

PW signaling protocols include LDP and BGP. They are responsible for creating and maintaining PWs.

- When using LDP sessions to set up PWs

When using LDP sessions to set up PWs, you should set up two types of LDP sessions: LDP sessions set up through the basic discovery mechanism and LDP sessions set up through the extended discovery mechanism. The former is used to set up public LSPs and the latter to transmit the label mapping messages of PWs. The setup and disconnection of a PW are as follows:

- PEs exchange Hello packets through the target LDP session (through the extended LDP discovery mechanism) and set up the LDP session.
- When the status of the AC at one PE end is Up, the PE assigns a label to the corresponding PW.
- The PE encodes the label value and the PW ID into the FEC TLV and sends a label mapping message to the peer PE through the target LDP session.

- Upon receipt of the label mapping message, the peer PE decodes the PW ID and label value and checks whether the interface parameters (such as MTU) and PW types are consistent.
- The PW is set up after both ends exchange their label values and verify the validity of PW IDs and interface parameters.
- To disconnect a PW, a PE sends a label withdrawal message to the peer PE. Then the PW is disconnected.

Caution

The following conditions must be met when you set up a PW through LDP; otherwise, the PW cannot be set up:

1) The MTUs and PW types on the devices at both ends of the PW must be consistent.

2) The PW IDs on the devices at both ends of the PW must be consistent.

- Using BGP signaling to set up PWs

Unlike using LDP signaling, this method does not require static configuration of the connection between CEs. Instead, the entire carrier network is divided into different VPNs and CEs are numbered globally in VPN sites. Similar to BGP/MPLS L3VPN, BGP signaling uses VPN Target to identify CE sites that belong to the same VPN. The process of discovering VPN sites by using VPN Target is called the automatic discovery. If BGP is used as the signaling protocol, there will be two stages. The first stage is automatic discovery and the second stage is to set up bidirectional PWs between PEs based on the result of the automatic discovery.

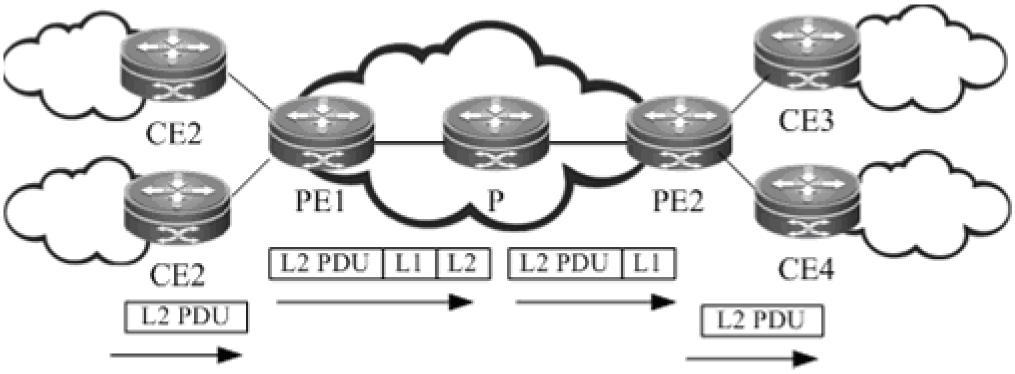
The method of establishing PWs by using BGP has brought in a concept of the label blocks. It uses the label block to allocate labels for multiple links at a time. Users can specify the site range for a local CE, indicating the number of remote CEs that can be connected with the CE. The system allocates one label block to the CE at a time. The label block's size is equal to the site range. In this way, users can allocate extra labels for VPN. This may cause waste of the label resource in a short term but can reduce the configuration workload for VPN expansion.

Given the BGP's characteristics, such as the route reflector's characteristics, this method of setting up PWs can reduce the full inter-connection of BGP sessions, facilitating the expansion of capacity.

Basic Forwarding Process

An L2VPN adopts the two-layer label stack to transmit services on the backbone network. The outer label is used to forward packets on the backbone network and the inner VC label is used to identify VC instances on PEs. Based on the inner VC label, a PE determines the CE to which the packets needs to be sent. The following figure shows the forwarding process.

Figure 48



After receiving a Layer 2 packet from CE2, PE1 searches for the PW forwarding entry based on the PW associated with the AC and learns that the next hop is PE2 and the PW label is L1. PE1 then searches for the public LSP based on the next hop (PE2) and obtains the outer label L2. As a result, PE1 encapsulates an MPLS header and pushes the inner label L1 and outer label L2 to the Layer packet and sends the packet to P. P forwards the MPLS packet based on the label and the penultimate hop popping (PHP) P pops out the outer label and sends the packet to PE2. Upon receipt of the packet, PE2 searches for the PW ID entry based on the inner label L1, learns the output interface (that is, the egress AC), pops out the inner label, and directly sends the Layer 2 packet to the destination CE4.

VCCV

Virtual circuit connectivity verification (VCCV) is a PW end-to-end connectivity fault detection and diagnosis mechanism. In brief, VCCV sends VCCV packets from the ingress node to the egress node of the PW to verify the end-to-end data plane connectivity of the PW and locate a connectivity fault in the PW. VCCV defines a series of messages to verify the connectivity of the PW. The VCCV packets must be encapsulated in the same way and pass the same tunnel as the PW packets to ensure that the VCCV packets are transmitted in the same path as the PW data packets. Therefore, VCCV needs to create a control channel that is associated with the PW and has the same path as the PW data if possible. The VCCV packets are transmitted in this control channel. The data in the control channel is not forwarded to the CE device, but is intercepted and processed on the PE device.

There are two VCCV modes: VCCV ping and VCCV traceroute. The VCCV ping mode can detect the PW connectivity. The VCCV traceroute mode can track the PW path to locate a connectivity fault in the PW.

VCCV Control Channel Type

VCCV packets are transmitted on the same PW bidirectional virtual circuit (VC) as PW service data packets, and have the same public network tunnel label and VC label encapsulated as common PW data packets. In addition, VCCV packets are transmitted in the same path as the PW data packets. The channel that transmits VCCV packets is called the control channel, and VCCV supports the following three types of control channels:

- Type 1: control word control channel.

VCCV packets that are transmitted in the control word control channel carry a control word. The first nibble of the control word carried in a VCCV packet is 0001b, indicating that this packet is a VCCV packet, which is different from the first nibble 0000b of the control word of a PW data packet. The control word can be used to distinguish the VCCV packet from common PW data packets and instruct the outbound PE router to process the VCCV packet.

- Type 2: router alert label control channel.

A control channel of this type is also referred to as router alert (RA) label control channel. An RA label is added to the PW label of a VCCV packet to force the outbound PE router to process the VCCV packet.

- Type 3: TTL timeout control channel.

For a single-segment PW, the time to live (TTL) of the PW label of a VCCV packet is set to 1. The TTL will expire on the outbound PE router, and as a result, the packet will be delivered to the CPU, which is forced to process the VCCV packet.

For a multi-segment pseudo wire (MS-PW), the TTL of the PW label of a VCCV packet is set according to the number of the PW segments. The TTL of the first segment is set to 1, the TTL of the second segment set to 2, and so on. The TTL will expire on the outbound PE router (S-PE or T-PE) in the PW segment, and as a result, the VCCV packet will be delivered to the CPU, which is forced to process the VCCV packet.

Caution Ruijie products do not support Type 1 and only support Types 2 and 3.

VCCV Ping

VCCV ping is used to verify the PW data plane connectivity. It is an extension of MPLS LSP ping. A PE device sends a VCCV ping request packet to the peer PE by running the user command. If no connectivity fault exists on the PW, the packet is forwarded to the egress PE device on the PW and delivered to the control plane. Otherwise, the packet will be discarded by the intermediate label switch router (LSR), which is also called transit LSR, or sent to its control plane. When receiving the packet on the control plane, the LSR verifies whether itself is the egress PE device of the PW based on the PW FEC information carried in the packet and responds to the source PE device. The VCCV response packet contains the relevant return code, which indicates the network status of the LSR. With VCCV ping, the source LSR possibly obtains a series of results based on the return code (indicating that the packet reaches the target PW egress and a response is returned successfully, the packet reaches a wrong PW egress, the PW path is disconnected, the transmission times out, and so on.), so as to check the PW connectivity.

VCCV Traceroute

VCCV traceroute is used to track the PW path and check the consistency between the control plane and data plane of all LSRs on the PW. VCCV traceroute is an extension of MPLS LSP traceroute.

VCCV traceroute also uses the MPLS echo request packets and response packets to track the PW path. In VCCV traceroute mode, multiple MPLS echo request packets are successively sent and the TTLs of the VC labels of these packets increase progressively. The TTL of the tunnel label of the first VCCV traceroute packet is 255, and the TTL value of the VC label is 1. The TTL of the VC label of each subsequent echo request packet increases by 1 compared with the previous one.

For a single-segment PW, if no connectivity fault exists on the PW, the packet is forwarded to the egress PE device on the PW and delivered to the control plane. The egress PE device verifies itself as the outbound PE device and directly returns a response packet as the egress PE device. Otherwise, the packet will be discarded by the intermediate LSR or sent to its control plane. When receiving the packet on the control plane, the LSR performs verification based on the information carried in the packet and responds to the source PE device.

For a MS-PW, if no connectivity fault exists on the MS-PW, the echo request packet sent in VCCV traceroute mode is forwarded to the intermediate S-PE device and the egress PE device and to the control plane. The intermediate S-PE device verifies the forwarding as an intermediate S-PE forwarding behavior and returns L to indicate that itself is the intermediate S-PE device on the MS-PW. The egress PE device verifies itself as the egress PE and returns a response packet as the egress PE device. Otherwise, the packet will be discarded by the intermediate LSR or sent to its control plane for processing. The intermediate LSR implements verification based on the information carried in the packet and then responds to the source PE device.

VPWS

Understanding VPWS

As an end-to-end bearer technology of Layer 2 services, VPWS is a P2P L2VPN. VPWS provides MPLS network-based L2VPN services so that carriers can provide L2VPN services based on various protocols including ATM, FR, VLAN, Ethernet and PPP over a unified MPLS network. Moreover, the MPLS network can provide traditional IP, MPLS L3VPN and other services. Simply speaking, MPLS L2VPN is to transmit users' layer-2 data transparently on the MPLS network.

- Signaling protocol of VPWS

VPWSs can be divided into Martini and Kompella VPWSs by signaling protocol. Martini VPWSs use LDP as the signaling protocol while Kompella VPWSs use BGP as the signaling protocol.

- Intercommunication of heterogeneous media of VPWSs

If CEs on two ends of the same L2VPN feature different link types, L2VPN's feature of heterogeneous media intercommunication are needed. According to the suggestion of draft-kompella-ppvpn-l2vpn, the encapsulation type of the L2VPN interface of PEs needs to be ip-interworking during the establishment of the L2VPN connection. Users' layer-3 data (IP packets) is transmitted on the MPLS network transparently.

Configuring Martini VPWS

Configuring a Public Tunnel

You must set up an LSP on the public network to carry VC services. To run MPLS on the backbone network, you must enable LDP on Ps and PEs to establish a public network tunnel. This means that you have to configure LDP on the routers and enable MPLS forwarding on each interface. The configuration procedure is as follows:

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **mpls ip**	Enables MPLS globally. ⚠ Caution This command is not applicable to switch chip forwarding.
Ruijie(config)# **mpls router ldp**	Enables LDP and enters MPLS routing configuration mode.
Ruijie(config-**mpls-router)# ldp router-id interface loopback** *id* [**force**]	Configures the LDP router ID. The IP address of the loopback interface is generally used as the router ID.
Ruijie(config-mpls-router)# **exit**	Exits MPLS routing configuration mode.
Ruijie(config)# **interface** *type ID*	Enters public network interface configuration mode.
Ruijie(config-**if**-*type ID*)# **ip address** *ip-address mask*	Assigns an IP address to the interface.
Ruijie(config-if-*type ID*)# **label-switching**	Enables MPLS forwarding on the interface at the public network side.
Ruijie(config-if-*type ID*)# **mpls ip**	Enables LDP on the interface.

Ruijie(config-if-*type ID*)# **ip ref**	☑ For routers, the fast forwarding function of the interface must be enabled. You do not need to use this command on switches.
Ruijie(config-if-*type ID*)# **show running-config**	Displays all configuration information.

# Configure an MPLS network.

```
Ruijie# configure terminal
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# )# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# exit
Ruijie(config)# interface gigabitethernet 1/1
```

# The **no switchport** command is used on switches to switch the port mode to routed port mode. It does not apply to routers and does not need to be used on routers.

```
Ruijie(config-if-gigabitethernet 1/1)# no switchport
```

# Enable the fast forwarding function of the interface on routers. You do not need to use this command on switches.

```
Ruijie(config-if-gigabitethernet 1/1)# no switchport
Ruijie(config-if-gigabitethernet 1/1)# ip address 192.168.10.1 255.255.255.0
Ruijie(config-if-gigabitethernet 1/1)# label-switching
Ruijie(config-if-gigabitethernet 1/1)# mpls ip
```

Configuring Remote LDP Peers

A PW is set up and maintained by the extended LDP. If other LSRs exist between two PEs, use the extended LDP discovery mechanism to set up a remote LDP session between the PEs and assign PW labels in the session. The procedures for configuring a remote LDP peer and setting up a remote LDP session are as follows:

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **mpls router ldp**	Enables LDP and enters LDP configuration mode.
Ruijie(config-mpls-router)# **neighbor** *ip-address*	Configures a remote LDP session.
Ruijie(config-mpls-router)# **show running-config**	Displays all configuration information.

# Configure a remote LDP peer at 3.3.3.3.

```
Ruijie# configure terminal
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# neighbor 3.3.3.3
Ruijie(config)# exit
```

⚠ Caution

The PW label messages of the LDP are not affected by the LDP label distribution mode or label retention mode. The LDP is forced to work in DU and liberal label retention mode.

Configuring User Access VPWS

VPWS Access Modes for Switches

Only Ethernet VPWS services are provided on switches. According to whether the packet carries a VLAN tag, user access can be divided into the following modes:

59) Access interface access
60) Trunk interface access
61) Dot1q Tunnel interface access

Caution

VPWS services are supported by only VLAN interfaces (that is, SVI interfaces on switches). In addition, the VLAN can have only one member interface. You have trouble in enabling both IP and VPWS services on VLAN interfaces at the same time.

One VLAN interface can be bound to only one VC instance. The same VC instance cannot be bound to different VLAN interfaces.

When the **xconnect** command is used to specify the neighbor address of the VC peer end, you must use the router ID of this peer end as the peer address. The router ID of this peer end must be the 32-bit address of the loopback interface.

After the port protection mode is enabled on the member ports of the AC end of L2VPN, the port protection mode does not take effect for non-Tunk member ports.

These access modes are described in details as follows:

- VLAN access interface access

This mode applies when user packets transmitted on ACs are not encapsulated through 802.1Q (that is, packets without VLAN tags). Use the following commands to configure VLAN access interface access in privileged mode.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **interface** *type ID*	Enters interface configuration mode.
Ruijie(config-if-*type ID*)# **switchport mode access**	Enables the interface to work in access mode.
Ruijie(config-if-*type ID*)# **switchport access vlan** *vlan-id*	Sets the interface as a member interface of a VLAN.
Ruijie(config-if-*type ID*)# **exit**	Exits interface configuration mode.
Ruijie(config)# **interface vlan** *vlan-id*	Creates and enters VLAN interface configuration mode.
Ruijie(config-if-*type ID*)# **xconnect** *vc_id vc_peer* **encapsulation mpls** {**ethernet|ethernetvlan**} **raw**	Creates a VC and configures the raw encapsulation mode.
Ruijie(config-if-*type ID*)# **show running-config**	Displays all configuration information.

# Configure **Gigabitethernet** *1/1* as an access interface and configure VPWS services for the corresponding VLAN interface.

```
Ruijie# configure terminal
Ruijie(config)# interface gigabitethernet 1/1
Ruijie(config-if-gigabitethernet 1/1)# switchport mode access
Ruijie(config-if-gigabitethernet 1/1)# switchport access vlan 2
Ruijie(config-if-gigabitethernet 1/1)# exit
Ruijie(config)# interface vlan 2
Ruijie(config-if-vlan 2)# xconnect 2  2.2.2.2 encapsulation mpls ethernet raw
```

Caution

In access interface access mode, we recommend setting the PW type to Ethernet, and the PW encapsulation mode must be raw.

■ VLAN trunk interface access

This mode applies to the transmission of VPWS services from multiple users on the same AC. The PE determines the VPWS services for user packets based on their VLAN tags to provide the multiplexing of access interfaces.

Use the following commands to configure VLAN trunk interface access in privileged mode.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **interface** *type ID*	Enters interface configuration mode.
Ruijie(config-if-*type ID*)# **switchport mode trunk**	Enables the interface to work in trunk mode.
Ruijie(config-if-*type ID*)# **switchport trunk allow vlan add** *vlan-list*	Enables the trunk link to allow VLAN traffic.
Ruijie(config-if-*type ID*)# **exit**	Exits the interface configuration mode.
Ruijie(config)# **interface vlan** *vlan-id*	Creates and enters VLAN interface configuration mode.
Ruijie(config-if-*type ID*)# **xconnect** *vc_peer vc_id* **encapsulation mpls ethernetvlan tagged**	Creates a VC and configures the tagged encapsulation mode.
Ruijie(config-if-*type ID*)# **show running-config**	Displays all configuration information.

# Configure **Gigabitethernet** *1/1* as a trunk port and configure VPWS services for the corresponding VLAN interface.

```
Ruijie# configure terminal
Ruijie(config)# interface gigabitethernet 1/1
Ruijie(config-if-gigabitethernet 1/1)# switchport mode trunk
Ruijie(config-if-gigabitethernet 1/1)# switchport trunk  allowed vlan add  2 3
Ruijie(config-if-gigabitethernet 1/1)# exit
Ruijie(config)# interface vlan 2
Ruijie(config-if-vlan 2)# xconenct 2  2.2.2.2 encapsulation mpls ethernetvlan tagged
Ruijie(config-if-vlan 2)# exit
Ruijie(config)# interface vlan 3
Ruijie(config-if-vlan 2)# xconenct 3  2.2.2.2 encapsulation mpls ethernetvlan tagged
```

Caution

In trunk interface access mode, we recommend setting the PW type to **ethernetvlan**, and the PW encapsulation mode must be tagged.

The L2VPN service cannot be bound to the Native VLAN of the Trunk interface.

- VLAN tunnel interface access

This mode applies when user service packets transmitted on ACs carry private VLAN tags if users access VPWS services. In this mode, the PE forwards all packets received from the VLAN tunnel interface without any changes. This mode requires the VLAN member interfaces that connect PEs with CEs to work in tunnel mode.

Use the following commands to configure VLAN tunnel interface access in privileged mode.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **interface** *type ID*	Enters interface configuration mode.
Ruijie(config-if-*type ID*)# **switchport access** *vlan-id*	Sets the interface as a VLAN member interface.
Ruijie(config-if-*type ID*)# **switchport mode dot1q-tunnel**	Sets the interface to work in tunnel mode.
Ruijie(config-if-*type ID*)# **exit**	Exits interface configuration mode.
Ruijie(config)# **interface** vlan *vlan-id*	Creates and enters VLAN interface configuration mode.
Ruijie(config-if-*type ID*)# **xconnect** *vc_id vc_peer* **encapsulation mpls ethernet raw**	Creates a VC and configures the raw encapsulation mode.
Ruijie(config-if-*type ID*)# **show running-config**	Displays all configuration information.

# Configure **Gigabitethernet** *1/1* as a VLAN tunnel interface and configure VPWS services for the corresponding VLAN interface.

```
Ruijie# configure terminal
Ruijie(config)# interface gigabitethernet 1/1
Ruijie(config-if-gigabitethernet 1/1)# switchport mode dot1q-tunnel
Ruijie(config-if-gigabitethernet 1/1)# switchport access 2
Ruijie(config-if-gigabitethernet 1/1)# exit
Ruijie(config)# interface vlan 2
Ruijie(config-if-vlan 2)# xconenct 2  2.2.2.2 encapsulation mpls ethernet raw
Ruijie(config-if-vlan 2)# exit
```

Caution

For the access mode of the VLAN tunnel port, we recommend setting the PW type to **ethernet** and the encapsulation mode must be raw.

For the access mode of the VLAN tunnel interface, only the basic QinQ is supported.

VPWS Access Modes for Routers

There are several ways for users to access VPWS services provided by routers. Users can choose access modes according to actual application needs. Services provided by the VPWS depend on the link protocol adopted by the interface that connects the PE with the CE. Currently, the following four point-to-point L2VPN services are supported:

1) Simulative Ethernet line service
2) Simulative 802.1Q line service
3) Simulative PPP line service
4) Simulative HDLC line service

For the PE, the four L2VPN line services correspond to four access modes.

⚠ Caution

You have trouble in enabling both IP and VPWS services on VLAN interfaces at the same time.

One VLAN interface can be bound to only one VC instance. The same VC instance cannot be bound to different VLAN interfaces.

☑ For the router product, VPWS services can be properly forwarded only when the fast forwarding function is enabled on the interfaces connected to the access user VPWS and to the public network.

⚠ Caution

When the **xconnect** command is used to specify the neighbor address of the VC peer end, you must use the router ID of this peer end as the peer address. The router ID of this peer end must be the 32-bit address of the loopback interface.

- Ethernet access

In this mode, the interface between a PE and a CE encapsulates the Ethernet link protocol and provides VPWS services. The CE connects to the PE through the Ethernet link and requests Ethernet frames transmitted transparently by the PE. The Ethernet interface access mode applies when user service packets transmitted on ACs carry private VLAN tags or do not carry VLAN tags in the case of VPWS service access. In this mode, all packets received by PEs from the interface are forwarded without any changes.

Use the following commands to configure Ethernet access in privileged mode.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **interface** *type ID*	Enters interface configuration mode.
Ruijie(config-if-*type ID*)# **xconnect** *vc_peer vc_id* **encapsulation mpls ethernet raw**	Creates a VC, and configure the Ethernet PW type and the raw encapsulation mode.
Ruijie(config-if-*type ID*)# **ip ref**	☑ For routers, the fast forwarding function of the interface must be enabled. You do not need to use this command on switches.
Ruijie(config-if-*type ID*)# **show running-config**	Displays all configuration information.

# Configure VPWS services for the Ethernet access interface **Gigabitethernet** *1/1*.

```
Ruijie# configure terminal
Ruijie(config)# interface gigabitethernet 0/1
```

# Enable the fast forwarding function of the interface on routers. You do not need to use this command on switches.

```
Ruijie(config-if-gigabitethernet 1/1)# ip ref
Ruijie(config-if-gigabitethernet 0/1)# xconnect 2.2.2.2 2 encapsulation mpls ethernet raw
Ruijie(config-if-gigabitethernet 0/1)# exit
```

- Ethernet sub-interface access

In this mode, the interface between a PE and a CE encapsulates the 802.1Q link protocol and provides VPWS services. The CE connects to the PE through the Ethernet sub-interface and requests Ethernet frames transmitted transparently by the PE. The Ethernet sub-interface access mode applies when several VPWS services for multiple users are transmitted on an access physical link. The PE matches packets with VPWS services according to dot1q tags carried by the user packets to provide the multiplexing of access interfaces. In this access mode, packets sent by CEs to PEs must carry VLAN tags.

Use the following commands to configure Ethernet sub-interface access in privileged mode.

Command	**Function**
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **interface** *type ID*	Enters interface configuration mode.
Ruijie(config-if-*type ID*)# **xconnect** *vc_peer vc_id* **encapsulation mpls ethernetvlan tagged**	Creates a VC, and configures the Ethernet VLAN PW type and the tagged encapsulation mode.
Ruijie(config-if-*type ID*)# **ip ref**	☑ For routers, the fast forwarding function of the interface must be enabled. You do not need to use this command on switches. Use the command to configure the sub-interface's fast forwarding function on its master interface.
Ruijie(config-**if**-*type ID*)# **show running-config**	Displays all configuration information.

# Configure **Gigabitethernet** *1/1* as an access interface and configure VPWS services for the corresponding VLAN interface.

```
Ruijie# configure terminal
Ruijie(config)# interface gigabitethernet 0/1.1
Ruijie(config-if-gigabitethernet 0/1.1)#encapsulation dot1Q 1
```

# Enable the fast forwarding function of the interface on routers. You do not need to use this command on switches.

```
Ruijie(config-if-gigabitethernet 1/1)# ip ref
Ruijie(config-if-gigabitethernet 0/1.1)# xconnect 2.2.2.2 2 encapsulation mpls ethernetvlan
tagged
Ruijie(config-if-gigabitethernet 0/1.1)# exit
```

⚠ Caution

If the VPWS service is enabled on both the master interface and sub-interface, or is enabled on one of them, packets without VLAN tags received by the Ethernet interface belong to the service provided by the master interface. If they carry VLAN tags and match with the sub-interface's VLAN ID, they belong to the service provided by the sub-interface.

To enable the sub-interface to support the fast forwarding function, use the **ip ref** command to enable the fast forwarding function on its master interface.

Users can use **mpls mtu** command to modify the **mpls mtu** value. By default, the value is equal to the **mtu** value of the interface.

- PPP access

In this mode, the interface that connects a PE with a CE encapsulates the PPP link protocol and provides VPWS services. The CE connects to the PE in PPP mode and requests the PPP frames transmitted transparently by the PE. The interface that encapsulates the PPP protocol is bound with the L2VPN service and its PE-end PPP protocol will be disabled. Therefore, the LCP and NCP of one CE will not interact with each other. The PE will transparently transmit the PPP negotiation controlling packet sent by CE. The PPP negotiation will be conducted between CEs.

Use the following commands to configure PPP access in privileged mode.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **interface** *type ID*	Enters interface configuration mode.
Ruijie(config-if-*type ID*)# **encapsulation ppp**	Enables the interface to encapsulate PPP.
Ruijie(config-if-*type ID*)# **xconnect** *vc_peer vc_id* **encapsulation mpls ppp**	Creates the VC and configures the PW type as PPP.
Ruijie(config-if-*type ID*)# **ip ref**	☑ For routers, the fast forwarding function of the interface must be enabled. You do not need to use this command on switches.
Ruijie(config-if-*type ID*)# **show running-config**	Displays the existing configuration information.

# Configure **serial** *1/0* to provide the VPWS service of transparently transmitting PPP frames.

```
Ruijie# configure terminal
Ruijie(config)# interface serial 1/0
Ruijie(config-if-serial 1/0)# encapsulation ppp
Ruijie(config-if-serial 1/0)# xconnect 2.2.2.2 2 encapsulation mpls ppp
Ruijie(config-if-serial 1/0)# exit
```

- HDLC access

In this mode, the interface that connects a PE with a CE encapsulates the HDLC link protocol and provides VPWS services. The CE connects to the PE in HDLC mode and requests the HDLC frames transmitted transparently by the PE.

Use the following commands to configure HDLC access in privileged mode.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **interface** *type ID*	Enters interface configuration mode.
Ruijie(config-if-*type ID*)# **encapsulation hdlc**	Enables the interface to encapsulate HDLC.
Ruijie(config-if-*type ID*)# **xconnect** *vc_peer vc_id* **encapsulation mpls hdlc**	Creates the VC and configures the PW type as HDLC.
Ruijie(config-if-*type ID*)# **show running-config**	Disables the existing configuration information.

# Configure **serial** *1/0* to provide the VPWS service of transparently transmitting HDLC.

```
Ruijie# configure terminal
Ruijie(config)# interface serial 1/0
```

# Enable the fast forwarding function of the interface on routers. You do not need to use this command on switches.

```
Ruijie(config-if-serial 1/0)# ip ref
Ruijie(config-if-serial 1/0)# encapsulation hdlc
Ruijie(config-if-serial 1/0)# xconnect 2.2.2.2 2 encapsulation mpls ppp
Ruijie(config-if-serial 1/0)# exit
```

Configuring Heterogeneous Media Communication VPWS

If CEs on two ends of the same L2VPN feature different link types, the L2VPN's feature of heterogeneous media intercommunication is needed. According to the suggestion of draft-kompella-ppvpn-l2vpn, the encapsulation type of the L2VPN interface of PEs needs to be ip-interworking during the establishment of the L2VPN connection. Users' IP packets are transmitted on the MPLS network transparently. When the L2VPN's heterogeneous media communication function is used, VPWS service interfaces of PEs on both ends must encapsulate ip-interworking. After PW connection is set up, packets are processed as follows:

62) After a PE receives packets from a CE and decapsulates the link layer, the PE transmits IP packets to the MPLS network.
63) IP packets are transmitted transparently through the MPLS network to the peer PE.
64) The peer PE re-encapsulates the IP packets according to its link layer protocol type and sends them to the CE connected to it.
65) The link layer control packet (such as PPP's IPCP) sent by the CE are processed by the PE without being transmitted on the MPLS network.
66) Non-IP packets (such as MPLS packets) are discarded and do not enter the MPLS network.

- Among the Ethernet interfaces on the PE, the following L2VPN interfaces can be encapsulated in ip-interworking mode:

67) Ethernet interface or sub-interface
68) GigabitEthernet interface or sub-interface

Note the following points:

69) After being encapsulated as ip-interworking, the PE's Ethernet interface processes only ARP and IP packets received by the local CE and discards the others including IPv6 packets.
70) When the PE receives IP packets from the CE, the dynamic MAC is not updated.
71) If the VPWS inbound interface that encapsulates ip-interworking on the PE receives the CE's ARP request packets, the VPWS inbound interface uses the PE's MAC address to reply regardless of the destination IP address.
72) Each Ethernet interface or sub-interface of the PE can be connected to only one CE and cannot be connected to multiple CEs or other devices through a hub or a layer-2 switch. Otherwise, MAC addresses learned by the PE will be covered, obstructing the forwarding.

- If a CE uses the PPP link protocol to access a PE, pay attention to the following points:

73) Unlike the negotiation that provides the homogeneous media L2VPN PPP simulative line service, the negotiation of PPP is conducted between the CE and the PE, rather than CEs; the address of the negotiation between PE and CE will not generate the corresponding route.
74) It supports PAP and CHAP authentication. The authentication method is the same as the common situation.
75) It does not support IPHC compression.

76) It supports transparent transmission of IP packets from the local CE to the peer CE and does not support transparent transmission though protocols such as MPLS and IPv6.

77) The **ppp ipcp address proxy** command must be used on the PE to specify the remote CE's IP address, which is the same as the IPCP proxy address of the PE as the remote CE. The address will not generate a route on the PE.

Use the following commands to configure the heterogeneous media communication VPWS access service in privileged mode.

Command	**Function**
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **interface** *type ID*	Enters interface configuration mode.
Ruijie(config-if-*type ID*)# **ppp ipcp address proxy** *remote-ce-ip-addr*	(Optional) When the interface that connects the PE to the CE encapsulates the PPP link protocol, configure the remote CE's IP address on the PE as the proxy address for the remote CE to conduct IPCP negotiation with the local CE.
Ruijie(config-if-*type ID*)# **xconnect** *vc_peer vc_id* **encapsulation mpls ip-interworking** [**local-ce mac** *mac*]	Creates VC and configures the PW type as the heterogeneous media communication. (Optional) **local-ce mac** *mac*: When the interface that connects PE to CE encapsulates the Ethernet protocol, the local CE's MAC address must be configured.
Ruijie(config-if-*type ID*)# **ip ref**	☑ For routers, the fast forwarding function of the interface must be enabled. You do not need to use this command on switches.
Ruijie(config-if-*type ID*)# **show running-config**	Displays the existing configuration information.

# Configure **serial** *1/0* to provide the heterogeneous media VPWS service of the PPP access mode.

```
Ruijie# configure terminal
Ruijie(config)# interface serial 1/0
Ruijie(config-if-serial 1/0)# encapsulation ppp
```

# Enable the fast forwarding function of the interface on routers. You do not need to use this command on switches.

```
Ruijie(config-if-gigabitethernet 1/1)# ip ref
Ruijie(config-if-serial 1/0)# ppp ipcp address proxy 192.168.1.1
Ruijie(config-if-serial 1/0)# xconnect 2.2.2.2 2 encapsulation mpls ip-interworking
Ruijie(config-if-serial 1/0)# exit
```

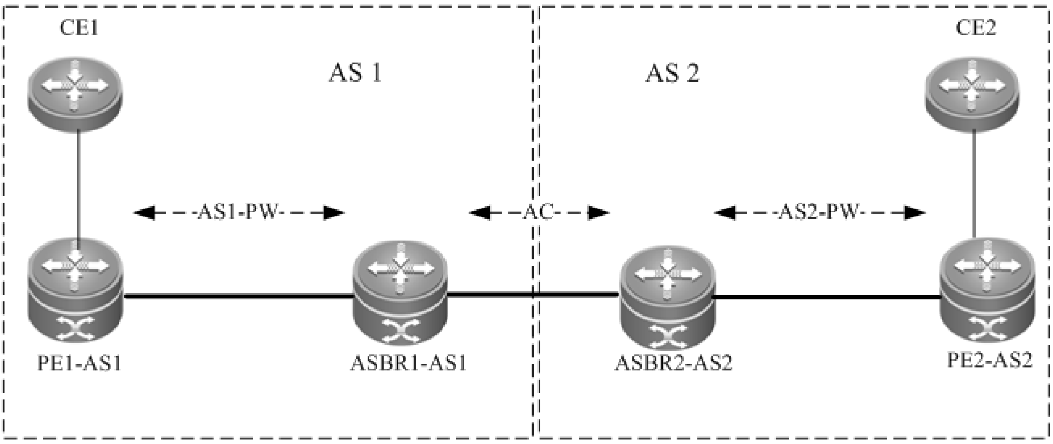
Configuring Inter-AS VPWS

There are two solutions for configuring Inter-AS Martini VPWS:

- Inter-AS Option A: This solution is simple and can be adopted when the number of inter-AS L2VPNs on ASBR is small.
- Inter-AS Option C: You do not need to create or maintain any VCs on ASBR. When each AS has numerous inter-AS L2VPNs, this solution can be applied to solve the bottleneck of the ASBR's scalability.

Inter-AS Option A

Figure 49 Option A Inter-AS VPWS



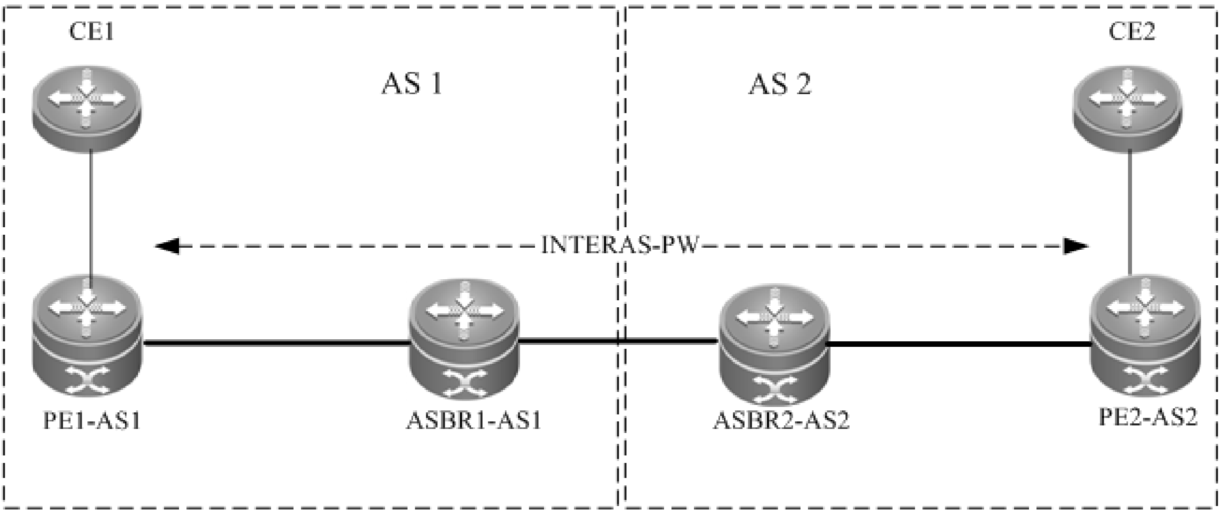
In the solution, ASBRs of two ASs are connected with each other and are PEs of their respective autonomous systems. Each ASBR considers the peer ASBR as its CE device. As shown in the preceding figure, for ASBR1-AS1 of AS1, ASBR2-AS2 of AS2 is only an accessed CE device; for ASBR2-AS2 of AS2, ASBR1-AS1 is also only an accessed CE device.

The Option A solution is easy to implement. You do not need to especially configure two PEs that serve as ASBRs or configure an IP address for the interface between ASBRs. For each inter-AS L2VPN or each Inter-AS PW, a logical or physical link must be allocated between ASBRs of two ASs. When there are numerous inter-AS PWs, great pressure is caused to ASBRs, hindering the expansion.

The solution's configuration is similar to the aforementioned basic VPWS configuration.

Inter-AS Option C

Figure 50 Option C Inter-AS VPWS



As shown in the preceding figure, Option C solution is to set up an inter-AS PW on two ASs directly and exchange PW tags. The principle is described as follows:

By sending tag IPv4 routes to the PE in respective ASs and sending tag IPv4 routes received by PEs in respective ASs to the ASBR peers of peer ASs, ASBRs connect the tunnel between two ASs and set up an LSP tunnel between the ingress PE and egress PE. Then, the inter-AS LDP remote session are set up between PEs in different ASs and PW information are exchanged.

In the solution, ASBRs do not need to maintain inter-AS L2VPN information or prepare a physical or logical interface for the inter-AS L2VPN. However, it needs to provide an MPLS tunnel. The L2VPN information is directly exchanged between PEs, decreasing the pressure on ASBRs and facilitating the scalability.

The configuration procedure is as follows:

78) Configure MPLS signaling.
79) Configure PEs.
80) Configure ASBRs.
81) Configure the remote LDP session.
82) Configure the user access VPWS.

- Configure MPLS signaling.

In each AS, enable the MPLS and LDP functions on PE and P devices and the interface that connects to the P or PE device in AS of the ABSR to set up a basic MPLS network. For the configuration procedures, see the chapter about basic MPLS configuration.

- Configuring PEs

Configure the PEs in ASs, set up an IBGP session between the PE and the AS, and exchange IPv4 routes that carry tags.

Command	Function
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **router bgp** *asn-number*	Configures BGP and enters BGP configuration mode.
Ruijie(config-router)# **neighbor** *asbr-address* **remote-as** *asbr-asn-number*	Sets up IBGP sessions between the PE and ASBR.
Ruijie(config-router)# **neighbor** *asbr-adddress* **update-sourc**e *interface-name*	Uses the loopback address as the source address of the BGP session set up between peers.
Ruijie(config-router)# **address-family ipv4**	Enters the IPv4 address family.
Ruijie(config-router-af)# **neighbor** *asbr-address* **send-label**	Enables IPv4 route tag switching.
Ruijie(config-router-af)# **show running-config**	Displays configuration information.

# Set up the IBGP session with the ASBR device 10.10.10.2 and enable the IPv4 route tag switching function.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 10.10.10.2 remote-as 1
Ruijie(config-router)# neighbor 10.10.10.2  update-source loopback 0
Ruijie(config-router)# address-family ipv4
Ruijie(config-router-af)# neighbor 10.10.10.2 activate
Ruijie(config-router-af)# neighbor 10.10.10.2 send-label
Ruijie(config-router-af)# exit
```

- Configuring ASBRs

Configure ASBRs to set up the IBGP session with the PE in the same AS and with the ASBR in the other AS. Enable the IPv4 route tag switching function on both sessions. Configure the PE address to be transmitted to another ASBR on the ASBR.

Command	Function
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **mpls ip**	Enables the device to support MPLS forwarding. This command is not applicable to switch chip forwarding.
Ruijie(config)# **mpls router ldp**	Enables LDP globally.
Ruijie(config-mpls-router)# **ldp router-id interface Loopback** *id* **force**	Configures the IP address of the loopback interface as the router ID.
Ruijie(config-mpls-router)# **advertise-labels for bgp-routes**	Allocates tags for BGP's route.
Ruijie(config-mpls-router)# **exit**	Returns to config mode.
Ruijie(config)# r**outer bgp** *asn-number*	Configures BGP and enters BGP configuration mode.
Ruijie(config-router)# **neighbor** *asbr-address* **remote-as** *asbr-asn-number*	Sets up EBGP session with ASBRs.
Ruijie(config-router)# **neighbor** *pe-address* **remote-as** *asn-number*	Sets up EBGP session with PEs.
Ruijie(config-router)# **neighbor** *pe-adddress* **update-source loopback** *id*	Uses the loopback address as the source address of the BGP session set up between PE peer.
Ruijie(config-router)# **address-family ipv4**	Enters the IPv4 address family.
Ruijie(config-router-af)# **network** *pe-address* **mask** *mask-value*	(Optional) Uses the **network** command to import the PE route received to BGP. The IGP protocol can be re-distributed to import the route.
Ruijie(config-router-af)# **neighbor** *asbr-address* **send-label**	Enables the IPv4 route tag switching function on BGP that is set up with the ASBR in another AS.
Ruijie(config-router-af)# **neighbor** *pe-address* **send-label**	Enables the IPv4 route tag switching function on BGP that is set up with the PE in the same AS.
Ruijie(config-router-af)# **neighbor** *asbr-ip-address* **route-map** *name* **out**	(Optional) Configures the route allocation policy. You can define route map rules (routemap) to control allocation of routes to neighbors and control whether these routes carry tags when they are sent.
Ruijie(config-router-af)# **neighbor** *asbr-ip-address* **route-map** *name* **in**	(Optional) Configures the route allocation policy. You can define route map rules (routemap) to control only routes that carry tags.
Ruijie(config-router-af)# **show running-config**	Displays configuration information.

# In the following example, the configured ASBR sets up an EBGP session with the ASBR in another AS (30.30.30.2) and the tag switching function is enabled for IPv4 routes. The IBG neighbor is created between the ASBR and the PE in the same AS (10.10.10.1) and the tag switching function is enabled for IPv4 routes.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 30.30.30.2 remote-as 2
Ruijie(config-router)# address-family ipv4
Ruijie(config-router-af)# neighbor 30.30.30.2 activate
Ruijie(config-router-af)# neighbor 30.30.30.2 send-label
Ruijie(config-router-af)# exit
Ruijie(config-router)# neighbor 10.10.10.1 remote-as 1
Ruijie(config-router)# neighbor 10.10.10.1 update-source loopback 0
Ruijie(config-router)# address-family ipv4
Ruijie(config-router-af)# neighbor 10.10.10.1 send-label
Ruijie(config-router-af)# exit
```

Caution

For the IBGP session set up between ASBR and the PE in the same AS, the **neighbor** *peer-address* **update-source** *loopback id* command must be used on the ASBR and PE to configure the address of the device's loopback interface as the source address of the session. Otherwise, the inter-AS LSP tunnel cannot be established.

The direct EBGP session established between ASBRs usually uses the direct connection interface's address as the source address of the BGP session to ensure that both ASBRs have routes that lead to each other. Therefore, you are not advised to use the **neighbor** *peer-address* **update-source** *loopback id* command on the EBGP session established based on direct connection to configure the address of the device's loopback interface as the source address of the session. If necessary, use the **neighbor ebgp-multihop** command to enable the multi-hop EBGP function, configure the static route on ASBR to ensure that the route can lead to the peer, and configure the static FTN to ensure that the inter-AS LSP's tunnel is available.

The **label-switching** command must be used on the interface between ASBRs to enable the interface's MPLS packet forwarding capability.

- Configuring the LDP remote session

Set up an inter-AS LDP remote session between PEs in two ASs. See the Configuring LDP Remote Peer chapter.

- Configuring the user access VPWS

For detailed configuration, see the Configuring User Access VPWS chapter.

- Checking the configuration result

Command	Function
show bgp ipv4 unicast labels	Displays label information allocated by BGP for IPv4 routes.
show mpls ldp neighbor	Displays neighbor information of LDP.
show mpls l2transport vc [**detail**]	Displays VC status information.

# View route and label information on an ASBR or a PE.

```
Ruijie # show bgp ipv4 unicast labels
Network         Next Hop        In Label/Out Label
1.1.1.1/32       192.167.1.1     17/18
1.1.1.2/32       192.167.1.1     nolabel/19
```

Field	Definition
Network	Route prefix

Nexthop	Route's next hop
In Label	Label (if any) allocated by the router
Out Label	Label (if any) learned from the next-hop router of the route

# View LDP session information on a PE.

```
Ruijie # show mpls ldp neighbor
Default VRF:
    Peer LDP Ident: 10.20.10.10:0; Local LDP Ident: 8.8.8.8:0
        TCP connection: 10.20.10.10.62488 - 8.8.8.8.646
        State: OPERATIONAL; Msgs sent/recv: 42/45; UNSOLICITED
        Up time: 00:33:49
        LDP discovery sources:
          Link Peer on GigabitEthernet 2/1, Src IP addr: 192.168.201.220
          Targeted Hello 8.8.8.8 -> 10.20.10.10
        Addresses bound to peer LDP Ident:
          10.20.10.10   192.168.201.220 192.168.198.1   10.5.0.1
```

Field	Definition
Peer LDP Ident	LDP identifier of the peer neighbor of the LDP session
Local LDP Identifier	Router's LDP identifier
TCP connection	TCP connection that supports the LDP session
State	Status of the LDP session
Msgs sent/recv	Quantity of LDP messages sent to/received by the session peer
UNSOLICITED and ONDEMAND	Label allocation mode

# View VC status information.

```
Ruijie # show mpls l2transport vc detail
Local interface : VLAN 2, AC state: up
Peer address: 192.168.0.1 ,VC ID: 2, VC status: up
VC type: vlan     VC mode:tagged
Group id: 0    MTU: 1500
Control Word not support
Output interface: VLAN 300 , imposed label stack {22 ,501 }
MPLS VC label:  local 22, remote 22
```

Field	Definition
Local interface	Local interface bound to VC
AC state	AC status, up or down
Peer address	Peer IP address
VC ID	VC's unique identifier
VC status	VC status, up or down
VC type	VC type
VC mode	VC mode, tagged or raw (only applies to the Ethernet access mode)
Group id	VC's local group ID
MTU	MTU of the locally configured VC

Field	Definition
Control Word	Whether the control word is supported
Output interface	Output interface on the public network used to transmit the VC process
imposed label stack	Added label stack
MPLS VC label	**Local** indicates the label locally allocated for the VC while **Remote** indicates the label allocated by the peer for the VC.

Configuring Kompella VPWS

Configuring the Public Network Tunnel

The LSP tunnel must be set up on the public network to carry the VC service. To run MPLS on the backbone network, LDP must be run on Ps and PEs to establish the public network tunnel. This includes configuring the label allocation protocol for MPLS devices and enabling the MPLS forwarding on each interface. The configuration procedure is as follows:

Command	Function
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie# **mpls ip**	Enables MPLS forwarding globally. ⚠ Caution This command is not applicable to switch chip forwarding.
Ruijie(config)# **mpls router ldp**	Enables LDP and enters LDP configuration mode.
Ruijie(config-mpls-router)# **ldp router-id interface loopback** *id* [**force**]	Configures LDP's router ID, which is usually the IP address of the loopback interface.
Ruijie(config-mpls-router)# **exit**	Exits LDP configuration mode.
Ruijie(config)# **interface** *type ID*	Enters public network interface configuration mode.
Ruijie(config-if-*type ID*)# **label-switching**	Enables the interface's MPLS forwarding function.
Ruijie(config-if-*type ID*)# **mpls ip**	Enables the interface's LDP function and MPLS forwarding function.
Ruijie(config-if-*type ID*)# **ip ref**	Enables the fast forwarding function for routers. You do not need to be enable the function for switches.
Ruijie(config-if-type ID)# **end**	Returns to privileged mode.
Ruijie# **copy running-config startup-config**	Saves configuration.

# Configure the public network tunnel between PEs.

```
Ruijie# configure terminal
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# exit
Ruijie(config)# interface gigabitethernet 1/1
Ruijie(config-if-GigabitEthernet 1/1)# label-switching
Ruijie(config-if-GigabitEthernet 1/1)# mpls ip
Ruijie(config-if-GigabitEthernet 1/1)# ip ref
Ruijie(config-if-GigabitEthernet 1/1)# end
Ruijie# copy running-config startup-config
```

Caution

The LDP is a topology-driven protocol. To ensure the normal working of the LDP, enable IPv4 routing protocols and ensure their normal operations.

Configuring the L2VPN Address Family

Kompella VPWS uses MP-BGP4 as the signaling protocol to transmit layer-2 information and VC labels, realizing point-to-point VPN. In addition, the MP-BGP4 protocol can be used as the auto-discovery protocol and connect remote CEs. The procedure for enabling the L2VPN address family is as follows:

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **router bg**p *asn-num*	Creates BGP and enters BGP configuration mode.
Ruijie(config)# **neighbor** *peer-address* remote-as *asn-number*	Sets up the IBGP session.
Ruijie(config)# **neighbor** *peer-adddress* **update-source** *interface-name*	Enables the IBGP session to use the address of the loopback interface as the session's source address.
Ruiije(config-router)# **address-family l2vpn vpws**	Enters the L2VPN VPWS address family.
Ruijie(config-router-af)# **neighbor** *ip-address* **activate**	Activates switching of L2VPN information on the BGP session.
Ruijie(config-router-af)# **neighbor** { *ip-address* | *peer-group-name* } **send-community** [**both | standard | extended**]	Specifies the extended community attribute to be sent to BGP neighbors.
Ruijie(config-router-af)# **end**	Returns from address family configuration mode to privileged mode.
Ruijie# **show bgp l2vpn vpws** { **all** [*id:offset* | **neighbor** *ip-address* **| summary**] }	Displays L2VPN address family information.

# Configure the L2VPN address family and enable VPWS information switching.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 100
Ruijie(config-router)# neighbor 10.10.10.1 remote-as 1
Ruijie(config-router)# neighbor 10.10.10.1 update-source loopback 0
Ruiije(config-router)# address-family l2vpn vpws
Ruijie(config-router-af)# neighbor 10.10.10.1 activate
Ruijie(config-router-af)# neighbor 10.10.10.1 send-community extended
Ruijie(config-router-af)# end
Ruijie# show bgp l2vpn vpws all
```

Configuring the Kompella VPWS Instance

The **l2 vfi** command can be used to create Kompella VPWS instances or enter Kompella VPWS configuration mode. The **no l2 vfi** command can be used to delete VFI instances. The unique local VFI instance name and the unique local device VPN ID must be specified when the instance is being created. The auto-discovery function must be enabled for the specified VFI instance. One VFI name must correspond to one VPN ID.

The **label-saving** command can be used to enable the label saving mode and allocate a label for the specified remote site. In label saving mode, **site range** does not take effect.

Command	**Function**
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **l2 vfi** *vpls-name* **vpnid** *vpn-id* **point-to-point**	Creates the Kompella VPWS instance and enters VPWS configuration mode.
Ruijie(config-vfi)# **rd** *rd_value*	Configures the RD value. Configure the RD first.
Ruijie(config-vfi)# **encapsulation mpls** [**ethernet** | **ethernetvlan | ppp | hdlc | ip-interworking**]	Specifies the encapsulation type of Kompella L2VPN PWs, which is Ethernet by default.
Ruijie(config-vfi)# r**oute-target** { **import | export | both** } *rt_value*	Configures RTs. Multiple RTs can be configured.
Ruijie(config-vfi)# **site-id** *id* [**site-range** *size*]	Configures the CE ID of a site and the site range. If the site range is not configured, the default value 16 will be adopted. Multiple site IDs can be configured for Kompella VPWS.
Ruijie(config-vfi-site)# **xconnect interface** *interface-type interface-number* **remote-ce-id** *id*	Binds the local interface and specifies the ID of the remote CE to be connected.
Ruijie(config-vfi-site)# **end**	Returns to privileged mode.
Ruijie# **copy running-config startup-config**	Saves configuration.

# Configure a Kompella VPWS instance.

```
Ruijie#configure terminal
Ruijie(config)# l2 vfi vpls-1 vpnid 1 point-to-point
Ruijie(config-vfi)# rd 100:1
Ruijie(config-vfi)# encapsulation mpls ethernet
Ruijie(config-vfi)# route-target both 4500:2
Ruijie(config-vfi)# site-id 1
Ruijie(config-vfi-site)# xconnect interface gi 2/2 remote-ce-id 2
Ruijie(config-vfi-site)#end
Ruijie# copy running-config startup-config
```

Caution

The **point-to-point** keyword must be specified after the **l2 vfi** command to create a Kompella VPWS instance.

VFI instances of one VPN must be configured with the same ID to facilitate management.

Each site's ID is globally unique in VFI instances.

VFI instances of all PEs on one VPN must be configured with the same VPWS PW type. Otherwise, BGP signaling cannot be used to establish PWs.

We recommend specifying the PW type as Ethernet VLAN for VPWS connected through the sub-interface and as Ethernet for VPWS connected through the Ethernet. If the interface type is different from the specified PW type, although PWs can be established, two PW types adopt different ways to process tags of user packets, causing problems to communication between CEs. VLAN tags carried by user packets are processed as p-tags in Ethernet VLAN PWs while are transmitted transparently as c-tags in Ethernet PWs.

Configuring User Access VPWS

A created VFI instance takes effect only after the user configured with the VFI instance is connected with the link. Multiple point-to-point VCs can be configured in one VFI.

Switch VPWS access mode

Switches can provide only Ethernet VPWS services. The user access mode can be divided into the following modes according to whether packets carry VLAN tags:

83) Access interface access
84) Trunk interface access
85) VLAN tunnel interface access

⚠ Caution

Only the VLAN interface (SVI interface of switches) can provide the VPWS service. IP and VPWS services cannot be enabled concurrently on the VLAN interface at the same time.

One VLAN interface can bind only one VC instance and one VC instance cannot be bound on different VLAN interfaces.

When the port protection mode is enabled on the AC-end member port of L2VPN, the port protection mode does not take effect on the member port if the corresponding member port is not a Trunk interface.

These access modes are described in details as follows:

- VLAN access interface access

This mode applies when user packets transmitted on ACs are not encapsulated by 802.1Q (that is, the packets do not carry VLAN tags). Use the following commands to configure VLAN access interface access in privileged mode.

Command	**Function**
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **interface** *type ID*	Enters interface configuration mode.
Ruijie(config-if-*type ID*)# **switchport mode access**	Sets the interface in access mode.
Ruijie(config-if-*type ID*)# **switchport access vlan** *vlan-id*	Sets the interface as a member port of a VLAN port.
Ruijie(config-if-*type ID*)# **exit**	Exits interface configuration mode.
Ruijie(config)# **l2 vfi** *vfi_name* **vpnid** *vpn_id* **point-to-point**	Enters VFI configuration mode.
Ruijie(config-vfi)**#** **rd** *rd_value*	Defines the RD value of the Kompella VPWS instance.
Ruijie(config-vfi)# **route-target** { **import | export | both** } *rt_value*	Configures RT.
Ruijie(config-vfi)# **encapsulation mpls ethernet**	Configures the encapsulation method of VFI as Ethernet.
Ruijie(config-vfi)# **site-id** *id* [**site-range** *size*]	Configures site information of the VFI.

Command	Function
Ruijie(config-vfi-site)#**xconnect interface vlan** *vlan-id* **remote-ce-id** *id*	Connects the interface that binds VFI locally with the VFI's remote CEs.

Caution

In VLAN access interface access mode, we recommend setting the PW encapsulation mode to **ethernet**, and the encapsulation modes on two ends of a PW must be the same.

# Configure **gigabitethernet** *1/1* to be connected through the access interface and configure VPWS services under the corresponding VLAN interface.

```
Ruijie# configure terminal
Ruijie(config)# interface gigabitethernet 1/1
Ruijie(config-if-gigabitethernet 1/1)# switchport mode access
Ruijie(config-if-gigabitethernet 1/1)# switchport access vlan 2
Ruijie(config-if-gigabitethernet 1/1)# exit
Ruijie(config)# l2 vfi vfiA vpnid 1 point-to-point
Ruijie(config-vfi)# rd 2:2
Ruijie(config-vfi)# route-target both 2:2
Ruijie(config-vfi)# encapsulation mpls ethernet
Ruijie(config-vfi)# site-id 1
Ruijie(config-vfi-site)# xconnect interface vlan 2  remote-ce-id 2
```

- VLAN trunk interface access

This mode applies when several VPWS services for multiple users are transmitted on an AC. PE devices can match packets with VPWS services according to VLAN tags carried by the user packets to provide the multiplexing of access interfaces.

Use the following commands to configure VLAN trunk interface access in privileged mode.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **interface** *type ID*	Enters interface configuration mode.
Ruijie(config-if-*type ID*)# **switchport mode trunk**	Sets the interface in trunk working mode.
Ruijie(config-if-*type ID*)# **switchport trunk allow vlan add** *vlan-list*	Sets the VLAN flows allowed to be transmitted on the trunk link.
Ruijie(config-if-*type ID*)# **exit**	Exits interface configuration mode.
Ruijie(config)# **l2 vfi** *vfi_name* **vpnid** *vpn_id* **point-to-point**	Enters VFI configuration mode.
Ruijie(config-vfi)# **rd** *rd_value*	Defines the RD value of the Kompella VPWS instance.
Ruijie(config-vfi)# **route-target** { **import | export| both** } *rt_value*	Configures RT.
Ruijie(config-vfi)# **encapsulation mpls ethernetvlan**	Configures the encapsulation method of VFI as Ethernet VLAN.
Ruijie(config-vfi)# **site-id** *id* [**site-range** *size*]	Configures site information of the VFI.

Command	Function
Ruijie(config-vfi-site)#**xconnect interface vlan** *vlan-id* **remote-ce-id** *id*	Connects the interface that binds VFI locally with the VFI's remote CEs.

Caution

In trunk access interface access mode, we recommend setting the PW encapsulation mode to **ethernetvlan**, and the encapsulation modes on two ends of a PW must be the same.
The L2 VPN service cannot be bound to the Native VLAN of the Trunk interface.

# Configure **gigabitethernet** *1/1* to be connected through the trunk interface and configure VPWS under the corresponding VLAN interface.

```
Ruijie# configure terminal
Ruijie(config)# interface gigabitethernet 1/1
Ruijie(config-if-gigabitethernet 1/1)# switchport mode trunk
Ruijie(config-if-gigabitethernet 1/1)# switchport trunk allowed vlan add 2 3
Ruijie(config-if-gigabitethernet 1/1)# exit
Ruijie(config)# l2 vfi vfiA vpnid 1 point-to-point
Ruijie(config-vfi)# rd 2:2
Ruijie(config-vfi)# route-target both 2:2
Ruijie(config-vfi)# encapsulation mpls ethernetvlan
Ruijie(config-vfi)# site-id 1
Ruijie(config-vfi-site)# xconnect interface vlan 2  remote-ce-id 2
```

- VLAN tunnel interface access

This mode applies when user service packets transmitted on an AC carry private VLAN tags if users access the VPWS service. In this mode, all packets received by PEs from the interface are forwarded without any changes. This mode requires the VLAN member ports between PEs with CEs to work in tunnel mode.

Use the following commands to configure VLAN tunnel interface access in privileged mode.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **interface** *type ID*	Enters interface configuration mode.
Ruijie(config-if-*type ID*)# **switchport access** *vlan-id*	Sets the VLAN member port of the interface.
Ruijie(config-if-*type ID*)# **switchport mode dot1q-tunne**l	Sets the interface to work in tunnel mode.
Ruijie(config-if-*type ID*)**# exit**	Exits interface configuration mode.
Ruijie(config)# **l2 vfi** *vfi_name* **vpnid** *vpn_id* **point-to-point**	Enters VFI configuration mode.
Ruijie(config-vfi)# **rd** *rd_value*	Defines the RD value of the Kompella VPWS instance.
Ruijie(config-vfi)# **route-target** { **import | export | both** } *rt_value*	Configures RT.
Ruijie(config-vfi)# **encapsulation mpls ethernetvlan**	Configures the encapsulation method of VFI as Ethernet VLAN.
Ruijie(config-vfi)# **site-id** *id* [**site-range** *size*]	Configures site information of the VFI.

Command	Function
Ruijie(config-vfi-site)#**xconnect interface vlan** *vlan-id* **remote-ce-id** *id*	Connects the interface that binds VFI locally with the VFI's remote CEs.

Caution

In VLAN tunnel interface access mode, we recommend setting the PW encapsulation mode to **ethernet**, and the encapsulation modes on two ends of a PW must be the same.
For the VLAN tunnel interface access mode, only the basic QinQ is supported.

# Configure **gigabitethernet** *1/1* to be connected through the VLAN tunnel interface and configure VPWS under the corresponding VLAN interface.

```
Ruijie# configure terminal
Ruijie(config)# interface gigabitethernet 1/1
Ruijie(config-if-gigabitethernet 1/1)# switchport mode dot1q-tunnel
Ruijie(config-if-gigabitethernet 1/1)# switchport access 2
Ruijie(config-if-gigabitethernet 1/1)# exit
Ruijie(config)# l2 vfi vfiA vpnid 1 point-to-point
Ruijie(config-vfi)# rd 2:2
Ruijie(config-vfi)# route-target both 2:2
Ruijie(config-vfi)# encapsulation mpls ethernetvlan
Ruijie(config-vfi)# site-id 1
Ruijie(config-vfi-site)# xconnect interface vlan 2  remote-ce-id 2
```

Router VPWS access mode

There are several ways for users to connect to VPWS services provided by routers. Users can choose access modes according to actual application needs. Services provided by the VPWS depend on the link protocol adopted by the interface that connects the PE with the CE. Currently, the following four point-to-point L2VPN services are supported:

86) Simulative Ethernet line service
87) Simulative 802.1Q line service
88) Simulative PPP line service
89) Simulative PPP line service

- Ethernet interface access

This mode applies when user service packets transmitted on ACs carry private VLAN tags or do not carry VLAN tags in the case of VFI service access. In this mode, all packets received by PEs from the interface are forwarded without any changes and the private tags are considered part of the data.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **interface** *type ID*	Enters interface configuration mode.
Ruijie(config-if-*type ID*)# **ip ref**	Enables fast forwarding of the Ethernet interface.
Ruijie(config-if-*type ID*)# **exit**	Exits interface configuration mode.
Ruijie(config)# **l2 vfi** *vfi_name* **vpnid** *vpn_id* **point-to-point**	Enters VFI configuration mode.

Ruijie(config-vfi)# **rd** *rd_value*	Defines the RD value of the Kompella VPWS instance.
Ruijie(config-vfi)# **route-target** { **import | export | both** } *rt_value*	Configures RT.
Ruijie(config-vfi)# **encapsulation mpls ethernet**	Configures the encapsulation mode of VFI as Ethernet.
Ruijie(config-vfi)# **site-id** *id* [**site-range** *size*]	Configures site information of the VFI.
Ruijie(config-vfi-site)# **xconnect interface** *type ID* **remote-ce-id** *id*	Connects the interface that binds VFI locally with the VFI's remote CEs.

# Configure **gigabitethernet** *1/0* to provide the VPWS service of transparently transmitting Ethernet frames.

```
Ruijie# configure terminal
Ruijie(config)# interface gigabitethernet 1/0
Ruijie(config-if-gigabitethernet 1/0)# ip ref
Ruijie(config-if-gigabitethernet 1/0)# exit
Ruijie(config)# l2 vfi vfiA vpnid 1 point-to-point
Ruijie(config-vfi)# rd 2:2
Ruijie(config-vfi)# route-target both 2:2
Ruijie(config-vfi)# encapsulation mpls ethernet
Ruijie(config-vfi)# site-id 1
Ruijie(config-vfi-site)# xconnect interface gigabitethernet 1/0 remote-ce-id 2
```

Caution

In Ethernet interface access mode, we recommend setting the PW encapsulation mode to **ethernet**, and the encapsulation modes on two ends of a PW must be the same.

The remote CE ID must be specified to establish the Kompella VPWS.

- Ethernet sub-interface access

In this mode, the interface between a PE and a CE is encapsulated by the 802.1Q link protocol and provides VPWS services. The CE connects to the PE through the Ethernet sub-interface and requests Ethernet frame transmitted transparently by the PE. The Ethernet sub-interface access mode applies when several VPWS services for multiple users are transmitted on a physical access link. PE devices can match packets with VPWS services according to dot1q tags carried by the user packets to provide the multiplexing of access interfaces. In this mode, packets sent by the CE to the PE must carry VLAN tags.

Use the following commands to configure Ethernet sub-interface access in privileged mode.

Command	**Function**
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **interface** *type ID*	Enters sub-interface configuration mode.
Ruijie(config-if-*type ID*)# **encapsulation dot1Q** *vlan-id*	Configures the VLAN ID to be encapsulated.
Ruijie(config-if-*type ID*)# **exit**	Exits interface configuration mode.
Ruijie(config)# **l2 vfi** *vfi_name* **vpnid** *vpn_id* **point-to-point**	Enters VFI configuration mode.
Ruijie(config-vfi)# **rd** *rd_value*	Defines the RD value of the Kompella VPWS instance.
Ruijie(config-vfi)# **route-target** { **import | export | both** } *rt_value*	Configures RT.

Ruijie(config-vfi)# **encapsulation mpls ethernetvlan**	Configures the encapsulation mode of VFI as Ethernet VLAN.
Ruijie(config-vfi)# **site-id** *id* [**site-range** *size*]	Configures site information of the VFI.
Ruijie(config-vfi-site)# **xconnect interface** *type ID* **remote-ce-id** *id*	Connects the interface that binds VFI locally with the VFI's remote CEs.

# Configure **gigabitethernet** *1/0.100* to provide the VPWS service of transparently transmitting Ethernet frames.

```
Ruijie# configure terminal
Ruijie(config)# interface gigabitethernet 1/0
Ruijie(config-if-gigabitethernet 1/0)# ip ref
Ruijie(config-if-gigabitethernet 1/0)# exit
Ruijie(config)# interface gigabitethernet 1/0.100
Ruijie(config-if-serial 1/0.100)# encapsulation dot1q 100
Ruijie(config-if-serial 1/0.100)# exit
Ruijie(config)# l2 vfi vfiA vpnid 1 point-to-point
Ruijie(config-vfi)# rd 2:2
Ruijie(config-vfi)# route-target both 2:2
Ruijie(config-vfi)# encapsulation mpls ethernetvlan
Ruijie(config-vfi)# site-id 1
Ruijie(config-vfi-site)# xconnect interface gigabitethernet 1/0.100  remote-ce-id 2
```

Caution

In sub-interface access mode, we recommend setting the PW encapsulation mode to **ethernetvaln**, and the encapsulation modes on two ends of a PW must be the same.

The **ip ref** command must be used to enable the fast forwarding function on the master interface of the sub-interface.

- PPP access

In this mode, the interface between a PE and a CE encapsulates the PPP link protocol and provides VPWS services. The CE connects to the PE in PPP mode and requests the PPP frames transmitted transparently by the PE. The interface that encapsulates the PPP protocol is bound with the L2VPN service and its PE-end PPP protocol will be disabled. Therefore, the LCP and NCP of one CE will not interact with each other. The PE will transparently transmit the PPP negotiation control packet sent by the CE. The PPP negotiation will be conducted between CEs.

Use the following commands to configure PPP access in privileged mode.

Command	**Function**
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **interface** *type ID*	Enters interface configuration mode.
Ruijie(config-if-*type ID*)# **ip ref**	Enables fast forwarding of the sub-interface.
Ruijie(config-if-*type ID*)# **encapsulation ppp**	Configures the PPP encapsulation protocol.
Ruijie(config-if-*type ID*)# **exit**	Exits interface configuration mode.
Ruijie(config)# **l2 vfi** *vfi_name* **vpnid** *vpn_id* **point-to-point**	Enters VFI configuration mode.
Ruijie(config-vfi)# **rd** *rd_value*	Defines the RD value of the Kompella VPWS instance.

Ruijie(config-vfi)# **route-target** { **import | export | both** } *rt_value*	Configures RT.
Ruijie(config-vfi)# **encapsulation mpls ppp**	Configures the encapsulation mode of VFI as PPP.
Ruijie(config-vfi)# **site-id** *id* [**site-range** *size*]	Configures site information of the VFI.
Ruijie(config-vfi-site)# **xconnect interface** *type ID* **remote-ce-id** *id*	Connects the interface that binds VFI locally with the VFI's remote CEs.

# Configure **serial** *1/0* to provide the VPWS service of transparently transmitting PPP frames.

```
Ruijie# configure terminal
Ruijie(config)# interface serial 1/0
Ruijie(config-if-serial 1/0)# encapsulation ppp
Ruijie(config-if-serial 1/0)# ip ref
Ruijie(config-if-serial 1/0)# exit
Ruijie(config)# l2 vfi vfiA vpnid 1 point-to-point
Ruijie(config-vfi)# rd 2:2
Ruijie(config-vfi)# route-target both 2:2
Ruijie(config-vfi)# encapsulation mpls ppp
Ruijie(config-vfi)# site-id 1
Ruijie(config-vfi-site)# xconnect interface serial 1/0 remote-ce-id 2
```

- HDLC access

In this mode, the interface between a PE and a CE encapsulates the HDLC link protocol and provides VPWS services. The CE connects to the PE in HDLC mode and requests the HDLC frames transmitted transparently by the PE.

Use the following commands to configure HDLC access in privileged mode.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **interface** *type ID*	Enters interface configuration mode.
Ruijie(config-if-*type ID*)# **ip ref**	Enables fast forwarding of the sub-interface.
Ruijie(config-if-*type ID*)# **encapsulation hdlc**	Configures the HDLC encapsulation protocol.
Ruijie(config-if-*type ID*)# **exit**	Exits interface configuration mode.
Ruijie(config)# **l2 vfi** *vfi_name* **vpnid** *vpn_id* **point-to-point**	Enters VFI configuration mode.
Ruijie(config-vfi)# **rd** *rd_value*	Defines the RD value of the Kompella VPWS instance.
Ruijie(config-vfi)# **route-target** { **import | export | both** } *rt_value*	Configures RT.
Ruijie(config-vfi)# **encapsulation mpls hdlc**	Configures the encapsulation mode of VFI as PPP.
Ruijie(config-vfi)# **site-id** *id* [**site-range** *size*]	Configures site information of the VFI.
Ruijie(config-vfi-site)# **xconnect interface** *type ID* **remote-ce-id** *id*	Connects the interface that binds VFI locally with the VFI's remote CEs.

# Configure **serial** *1/0* to provide the VPWS service of transparently transmitting HDLC frames.

```
Ruijie# configure terminal
Ruijie(config)# interface serial 1/0
Ruijie(config-if-serial 1/0)# encapsulation hdlc
Ruijie(config-if-serial 1/0)# ip ref
Ruijie(config-if-serial 1/0)# exit
Ruijie(config)# l2 vfi vfiA vpnid 1 point-to-point
Ruijie(config-vfi)# rd 2:2
Ruijie(config-vfi)# route-target both 2:2
Ruijie(config-vfi)# encapsulation mpls hdlc
Ruijie(config-vfi)# site-id 1
Ruijie(config-vfi-site)# xconnect interface serial 1/0  remote-ce-id 2
```

Configuring Heterogeneous Media Communication VPWS

If CEs on two ends of the same L2VPN feature different link types, the L2VPN's feature of heterogeneous media intercommunication is needed. According to the suggestion of draft-kompella-ppvpn-l2vpn, the encapsulation type of the L2VPN interface of PEs needs to be ip-interworking during the establishment of L2VPN connection. Users' IP packets are transmitted on the MPLS network transparently. When the L2VPN's heterogeneous media communication function is used, VPWS service interfaces of PEs on both ends must encapsulate ip-interworking. After PW connection is set up, packets are processed as follows:

90) After a PE receives packets from a CE and decapsulates the link layer, the PE transmits IP packets to the MPLS network.
91) IP packets are transmitted transparently through the MPLS network to the peer PE.
92) The peer PE re-encapsulates the IP packets according to its link layer protocol type and sends them to the CE connected to it.
93) The link layer control packet (such PPP's IPCP) sent by the CE are processed by the PE without being transmitted on the MPLS network.
94) Non-IP packets (such as MPLS packets) are discarded and do not enter the MPLS network.

■ Among the Ethernet interfaces on the PE, the following L2VPN interfaces can be encapsulated in ip-interworking mode:

95) Ethernet interface or sub-interface
96) GigabitEthernet interface or sub-interface

Note the following points:

97) After being encapsulated as ip-interworking, the PE's Ethernet interface processes only ARP and IP packets received by the local CE and discards the others including IPv6 packets.
98) When the PE receives IP packets from the CE, the dynamic MAC is not updated;
99) If the VPWS inbound interface that encapsulates ip-interworking on the PE receives the CE's ARP request packets, the VPWS inbound interface uses the PE's MAC address to reply regardless of the destination IP address.
100) Each Ethernet interface or sub-interface of the PE can be connected to only one CE and cannot be connected to multiple CEs or other devices through a hub or a layer-2 switch. Otherwise, MAC addresses learned by the PE will be covered, obstructing the forwarding.

■ If a CE uses the PPP link protocol to access a PE, pay attention to the following points:

101) Unlike the negotiation that provides the homogeneous media L2VPN PPP simulative line service, the negotiation of PPP is conducted between the CE and the PE, rather than CEs; the address of the negotiation between PE and CE will not generate the corresponding route.
102) It supports PAP and CHAP authentication. The authentication method is the same as the common situation.
103) It does not support IPHC compression.
104) It supports transparent transmission of IP packets from the local CE to the peer CE and does not support transparent transmission though protocols such as MPLS and IPv6.

105) The **ppp ipcp address proxy** command must be used on the PE to specify the remote CE's IP address, which is the same as the IPCP proxy address of the PE as the remote CE. The address will not generate a route on the PE.

Caution

If both ends of the CE is configured with the address of the same network segment, they can be connected no matter which IP address is configured for the PE-end PPP proxy as the CE has a network segment route that leads to the peer CE and the next-hop output interface can be obtained.

If two ends of the CE are configured with addresses of different network segments, the PE-end PPP proxy must be specified as the peer CE address and a route that leads to the peer CE's network segment must be configured statically in the CE.

Use the following commands to configure the heterogeneous media communication VPWS access service in privileged mode.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **l2 vfi** *vpws-name* **vpnid** *vpn-id* point-to-point	Creates the Kompella VPWS instance and enters VPWS configuration mode.
Ruijie(config-vfi)# **rd** *rd_value*	Configures the RD value. Configure the RD first.
Ruijie(config-vfi)# **encapsulation mpls ip-interworking**	Specifies ip-interworking as the encapsulation type of Kompella L2VPN PW.
Ruijie(config-vfi)# **route-target** { **import | export | both** } *rt_value*	Configures RTs. Multiple RTs can be configured.
Ruijie(config-vfi-site)# **xconnect interface** *interface-type interface-number* **remote-ce-id** *id*	Configures the local interface to be bound and specifies the ID of the CE to be connected remotely.
Ruijie(config-vfi-site)# **local-ce mac** *mac*	Specifies the CE's static MAC address.
Ruijie(config-vfi-site)# **exit-site-mode**	Returns to VFI configuration mode.
Ruijie(config-vfi)# **exit**	Returns to global configuration mode.
Ruijie(config)# **interface** *type ID*	Enters interface configuration mode.
Ruijie(config-if-*type ID*)# **ppp ipcp address proxy** *remote-ce-ip-addr*	(Optional) When the interface that connects the PE to the CE encapsulates the PPP link protocol, configure the remote CE's IP address on the PE as the proxy address for the remote CE to conduct IPCP negotiation with the local CE.
Ruijie(config-if-*type ID*)# **ip ref**	☑ For routers, the fast forwarding function of the interface must be enabled. You do not need to use this command on switches.
Ruijie(config-if-*type ID*)# **show running-config**	Displays the existing configuration information.

# Configure **serial** *1/0* to provide the heterogeneous media VPWS service of the PPP access mode.

```
Ruijie(config)# interface serial 1/0
Ruijie(config-if-serial 1/0)# encapsulation ppp
```

# Enable the fast forwarding function of the interface on routers. You do not need to use this command on switches.

```
Ruijie(config-if-serial 1/0)# ip ref
Ruijie(config-if-serial 1/0)# ppp ipcp address proxy 192.168.1.1
Ruijie(config-if-serial 1/0)# exit

Ruijie# configure terminal
Ruijie# l2 vfi vpws1 vpnid 1 point-to-point
Ruijie(config-vfi)# rd 100:1
Ruijie(config-vfi)# encapsulation mpls ip-interworking
Ruijie(config-vfi)# route-target both 100:1
Ruijie(config-vfi)# site-id 1
Ruijie(config-vfi-site)# xconnect interface serial 1/0 remote-ce-id 2
```

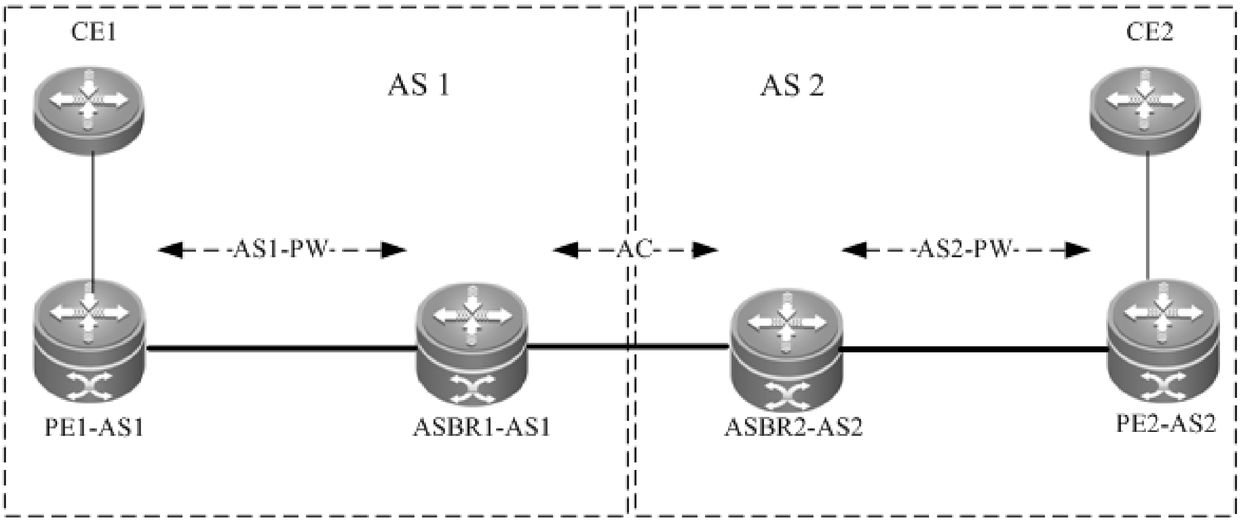
Configuring Inter-AS VPWS

In real application, multiple sites of a user's VPN may be connected to multiple SPs using different ASs or multiple ASs of an SP. The application mode in which the VPN crosses multiple ASs is called the inter-AS VPN. There are two solutions for configuring inter-AS VPWS:

- Inter-AS Option A: This solution is simple and can be adopted when the number of inter-AS L2VPNs on ASBR is small.
- Inter-AS Option C: You do not need to create or maintain any VCs on ASBR. When each AS has numerous inter-AS L2VPNs, this solution can be applied to solve the bottleneck of the ASBR's scalability.

Option A

Figure 51 Option A Inter-AS VPWS



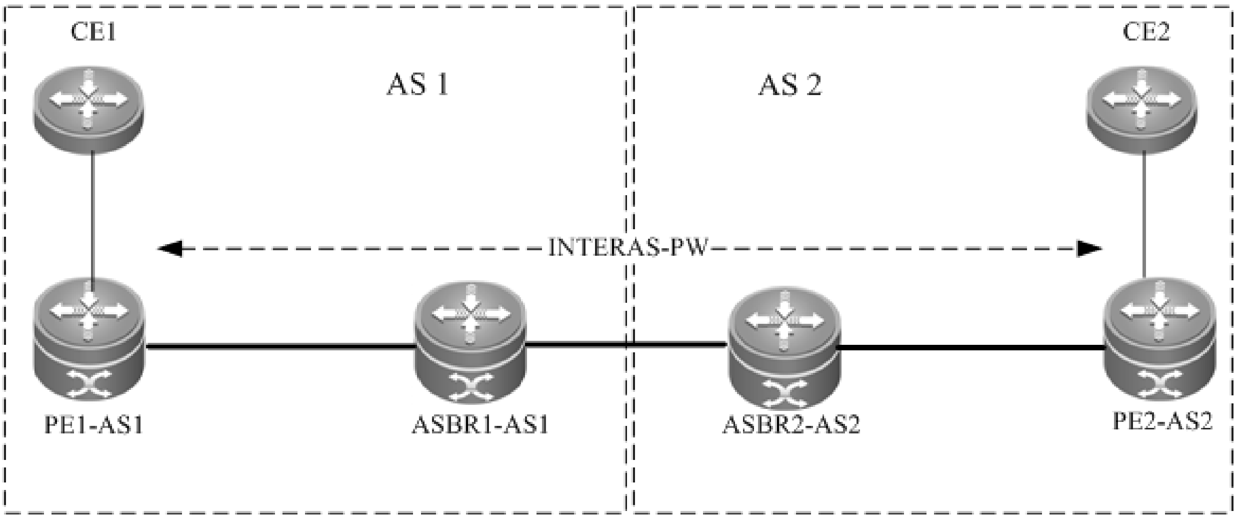
In the solution, ASBRs of two ASs are connected with each other and are PEs of their respective autonomous systems. Each ASBR considers the peer ASBR as its CE device. As shown in the preceding figure, for ASBR1-AS1 of AS1, ASBR2-AS2 of AS2 is only an accessed CE device; for ASBR2-AS2 of AS2, ASBR1-AS1 is also only an accessed CE device.

Option A is easy to implement. You do not need to especially configure two PEs that serve as ASBRs or configure an IP address for the interface between ASBRs. For each inter-AS L2VPN or each Inter-AS PW, a logical or physical link must be allocated between ASBRs of two ASs. When there are numerous inter-AS PWs, great pressure is caused to ASBRs, hindering the expansion.

The solution's configuration is similar to the aforementioned basic VPWS configuration.

Option C

Figure 52 Option C Inter-AS VPWS



As shown in the preceding figure, Option C is to set up an inter-AS PW on two ASs directly and switch PW labels. The principle is described as follows:

By sending tag IPv4 routes to the PE in respective ASs and sending tag IPv4 routes received by PEs in respective ASs to the ASBR peers of peer ASs, ASBRs connect the tunnel between two ASs and set up an LSP tunnel between the ingress PE and egress PE. Then, the Inter-AS LDP remote session are set up between PEs in different ASs and PW information are switched.

In the solution, ASBRs do not need to maintain inter-AS L2VPN information or prepare a physical or logical interface for the inter-AS L2VPN. However, it needs to provide an MPLS tunnel. The L2VPN information is directly switched between PEs, decreasing the pressure on ASBRs and facilitating the scalability.

The configuration procedure is as follows:

106) Configure MPLS signaling.
107) Configure PEs.
108) Configure ASBR.
109) Configure the L2VPN address family.
110) Configure a Kompella VPWS instance.
111) Configure the user access VPWS.
112) Check the configuration result.

- Configure MPLS signaling.

In each AS, enable the MPLS and LDP functions on PE and P devices and the interface that connects to the P or PE device in AS of the ABSR to set up a basic MPLS network. For the configuration procedures, see the chapter about basic MPLS configuration.

- Configuring PEs

Configure the PEs in ASs, set up an IBGP session between the PE and the AS, and exchange IPv4 routes that carry tags.

Command	Function
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **router bgp** *asn-number*	Configures BGP and enters BGP configuration mode.
Ruijie(config-router)# **neighbor** *asbr-address* **remote-as** *asbr-asn-number*	Sets up IBGP sessions between the PE and ASBR.
Ruijie(config-router)# **neighbor** *asbr-adddress* **update-source** *interface-name*	Uses the loopback address as the source address of the BGP session set up between peers.
Ruijie(config-router)# **address-family ipv4**	Enters the IPv4 address family.
Ruijie(config-router-af)# **neighbor** *asbr-address* **send-label**	Enables IPv4 route tag switching.
Ruijie(config-router-af)# **show running-config**	Displays configuration information.

# Set up the IBGP session with the ASBR device 10.10.10.2 and enable the IPv4 route tag switching function.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 10.10.10.2 remote-as 1
Ruijie(config-router)# neighbor 10.10.10.2  update-source loopback 0
Ruijie(config-router)# address-family ipv4
Ruijie(config-router-af)# neighbor 10.10.10.2 activate
Ruijie(config-router-af)# neighbor 10.10.10.2 send-label
Ruijie(config-router-af)# exit
```

- Configuring ASBRs

Configure ASBRs to set up the IBGP session with the PE in the same AS and with the ASBR in the other AS. Enable the IPv4 route tag switching function on both sessions and enable MPLS globally. Configure the PE address to be transmitted to another ASBR on the ASBR.

Command	Function
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **mpls ip**	Enables the device to support MPLS forwarding. ⚠ Caution This command is not applicable to switch chip forwarding.
Ruijie(config)# **mpls router ldp**	Enables LDP protocol globally.
Ruijie(config-mpls-router)# **ldp router-id interface loopbac**k *id* **force**	Configures the IP address of the loopback interface as the router ID.
Ruijie(config-mpls-router)# **advertise-labels for bgp-routes**	Allocates labels for BGP's route.
Ruijie(config-mpls-router)# **exit**	Returns to config mode.
Ruijie(config)# **router bgp** *asn-number*	Configures BGP and enters BGP configuration mode.
Ruijie(config-router)# **neighbo**r *asbr-address* **remote-as** *asbr-asn-number*	Sets up EBGP session with ASBRs.
Ruijie(config-router)# **neighbo**r *pe-address* **remote-as** *asn-number*	Sets up EBGP session with PEs.

Ruijie(config-router)# **neighbor** *pe-adddress* **update-source loopback** *id*	Uses the loopback address as the source address of the BGP session set up between PE peer.
Ruijie(config-router)# **address-family ipv4**	Enters the IPv4 address family.
Ruijie(config-router-af)# **network** *pe-address* **mask** *mask-value*	(Optional) Uses the **network** command to import the PE route received to BGP. The IGP protocol can be re-distributed to import the route
Ruijie(config-router-af)# **neighbor** *asbr-address* **send-label**	Enables the IPv4 route label switching function on BGP that is set up with the ASBR in another AS.
Ruijie(config-router-af)# **neighbor** *pe-address* **send-label**	Enables the IPv4 route label switching function on BGP that is set up with the PE in the same AS.
Ruijie(config-router-af)# **neighbor** *asbr-ip-address* **route-map** *name* **out**	(Optional) Configures the route allocation policy. You can define route map rules (routemap) to control allocation of routes to neighbors and control whether these routes carry labels when they are sent.
Ruijie(config-router-af)# **neighbor** *asbr-ip-address* **route-ma**p *name* **in**	(Optional) Configures the route allocation policy. You can define route map rules (routemap) to control only routes that carry labels.
Ruijie(config-router-af)# **show running-config**	Displays configuration information.

# In the following example, the configured ASBR sets up an EBGP session with the ASBR in another AS (30.30.30.2) and the label switching function is enabled for IPv4 routes. The IBG neighbor is created between the ASBR and the PE in the same AS (10.10.10.1) and the label switching function is enabled for IPv4 routes.

```
Ruijie# configure terminal
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# advertise-labels for bgp-routes
Ruijie(config-mpls-router)# exit
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 30.30.30.2 remote-as 2
Ruijie(config-router)# address-family ipv4
Ruijie(config-router-af)# neighbor 30.30.30.2 activate
Ruijie(config-router-af)# neighbor 30.30.30.2 send-label
Ruijie(config-router-af)# exit
Ruijie(config-router)# neighbor 10.10.10.1 remote-as 1
Ruijie(config-router)# neighbor 10.10.10.1 update-source loopback 0
Ruijie(config-router)# address-family ipv4
Ruijie(config-router-af)# neighbor 10.10.10.1 send-label
Ruijie(config-router-af)# exit
```

Caution

For the IBGP session set up between ASBR and the PE in the same AS, the **neighbor** *peer-address* **update-source** *loopback id* command must be used on the ASBR and PE to configure the address of the device's loopback interface as the source address of the session. Otherwise, the inter-AS LSP tunnel cannot be established.

The direct EBGP session established between ASBRs usually uses the direct connection interface's address as the source address of the BGP session to ensure that both ASBRs have routes that lead to each other. Therefore, you are not advised to use the **neighbor** *peer-address* **update-source** *loopback id* command on the EBGP session established based on direct connection to configure the address of the device's loopback interface as the source address of the session. If necessary, use the **neighbor ebgp-multihop** command to enable the multi-hop EBGP function, configure the static route on ASBR to ensure that the route can lead to the peer, and configure the static FTN to ensure that the inter-AS LSP's tunnel is available.

The **label-switching** command must be used on the interface between ASBRs to enable the interface's MPLS packet forwarding capability.

- Configuring the L2VPN address family

Configure L2VPN address families between PEs of ASs. For detailed configuration, see the Configuring L2VPN Address Family chapter.

- Configuring a Kompella VPWS instance

Configure L2VPN address families between PEs of ASs. For detailed configuration, see the Configuring Kompella VPWS Instance chapter.

- Configuring the user access VPWS

For detailed configuration, see the Configuring User Access VPWS chapter.

- Checking the configuration result

Command	**Function**
show bgp ipv4 unicast labels	Displays label information allocated by BGP for IPv4 routes.
show bgp l2vpn vpws all connections	Displays Kompella VPWS connection information.
show mpls l2transport vc [**detail**]	Displays VC status information.

# View route and label information on an ASBR or a PE.

```
Ruijie # show bgp ipv4 unicast labels
Network         Next Hop        In Label/Out Label
1.1.1.1/32      192.167.1.1     17/18
1.1.1.2/32      192.167.1.1     nolabel/19
```

Field	**Definition**
Network	Route prefix
Nexthop	Route's next-hop
In Label	Label (if any) allocated by the router
Out Label	Label (if any) learned from the next-hop router of the route

# View VC status information.

```
Ruijie # show mpls l2transport vc detail
Local interface : VLAN 2, AC state: up
Peer address: 192.168.0.1 ,VC ID: 2, VC status: up
```

```
VC type: vlan     VC mode:tagged
Group id: 0    MTU: 1500
Control Word not support
Output interface: VLAN 300 , imposed label stack {22 ,501 }
MPLS VC label:  local 22, remote 22
```

Field	**Definition**
Local interface	Local interface bound by VC
AC state	AC status, up or down
Peer address	VC's peer IP address
VC ID	VC's unique identifier
VC status	VC status, up or down
VC type	VC type
VC mode	VC mode, tagged or raw (only applies to the Ethernet access mode)
Group id	VC's local group ID
MTU	MTU of the locally configured VC
Control Word	Whether the control word is supported
Output interface	Output interface on the public network used to transmit the VC process
imposed label stack	Added label stack
MPLS VC label	Local and peer private labels bound for the VC

Other Parameters for Configuring Kompella VPWS

Configuring VPWS Instance Descriptors (Optional)

You can configure the descriptive information of each VPWS instance.

Command	**Function**
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **l2 vfi** *vfi-name*	Enters VFI configuration mode.
Ruijie(config-vfi)# **description** *vfi-description*	(Optional) Configures VFI's descriptive information.
Ruijie(config-vfi)# **end**	Returns to privileged mode.
Ruijie# **copy running-config startup-config**	Saves configuration.

Configuring VPWS's Compatibility (Optional)

By default, the PW's mtu value provided by L2VPN is 1500 bytes. If the same PW's mtu values on two PEs are different, PW connection cannot be set up between the two PEs. Some manufacturers' devices do not support configuring mtu in L2VPN instances. When such devices perform Kompella communication with devices of other manufacturers, the **ingore match l2-extcommunity** command can be used to ignore received mtu and matching detection of **EncapsType** and **Control Flag**, ensuring that the VC link is UP.

Command	**Function**
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **l2 vfi** *vpls-name* **vpnid** *vpn-id* **autodiscovery**	Creates the Kompella VPWS instance and enters VPWS configuration mode.

Ruijie(config-vfi)# **ingore match l2-extcommunity**	Configures ignoring detection of L2VPN expanded community attribute members.
Ruijie# **copy running-config startup-config**	Saves configuration.

Caution

This command only takes effect on Kompella L2VPN.

Configuring mtu of a VPWS Instance (Optional)

You can configure each VFI instance's mtu value, which is **1500** by default. The mtu value of VFI indicates the length of the packet that can be transmitted by the PW, or the length of the user's layer-2 packet plus the length of the PW-encapsulated packet. By default, if the PW does not enable the control word, assuming that two labels are encapsulated, the length of an Ethernet packet that can be transmitted is 1492 bytes, of which 8 bytes are encapsulated by the PW (2 labels).

Command	Function
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **l2 vfi** *name*	Enters VFI configuration mode.
Ruijie(config-vfi)# **mtu** *mtu*	(Optional) Configures the mtu value of the VFI.
Ruijie(config-vfi)# **end**	Returns to privileged mode.
Ruijie# **copy running-config startup-config**	Saves configuration.

Caution

The mtu values of one VFI instance on different PEs must be the same. Otherwise, the signaling protocol cannot establish PWs.

If the PW signaling protocol negotiation's mtu is modified, the mtu of the user access service interface must be adjusted (generally adjusted to the PW mtu length minus the encapsulated length); the PW's public-network-end output interface's mtu, MPLS mtu and PW mtu must be the same to ensure proper forwarding. The **mtu** command can be used on an interface to modify the interface's mtu. Use the **mpls mtu** command to modify the interface's MPLS mtu.

Configuring Static VC FTN and ILM Entries (Optional)

There are two methods to add VC FTN and ILM entries: by switching VC labels through the extended LDP and static configurations. These two methods cannot be applied to a VC instance at the same time. That is, if you add a VC FTN or ILM entry through the extended LDP, you cannot use the CLI command to add entries for the VC. Similarly, if you use the CLI command to add a VC FTN or ILM entry, you cannot add VC FTN or ILM entry by using the VPWS signaling protocol.

The procedure for configuring a VC FTN entry is as follows:

Command	Function
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **mpls static l2vc-ftn** *vc-id vc-peer-ip* **out-label** *out-label*	Configures an FTN entry for a static VC instance.
Ruijie(config)# **show running-config**	Displays all configuration information.

The procedure for configure a VC ILM entry is as follows:

Command	Function
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **mpls static ilm in-label** *in_label* **forward-action pop-l2vc-destport** *vc-id vc_peer_addr*	Configures an ILM entry for a static VC instance.
Ruijie(config)# **show running-config**	Displays all configuration information.

Caution

We do not recommend configuring static VC FTN and ILM when using the VPW dynamic signaling protocol to establish PW. For configured static VC forwarding table entries, the static public network LSP must be configured. VC is only effective when the corresponding public network LSP tunnel exists.
Static VC FTN and ILMF are not needed for heterogeneous media communication.
Static VC FTN and ILM do not take effect on a Kompella VPWS VC instance but only on a Martini VC instance. That is, they are generated by static forwarding table entries, not by LDP signaling.

Configuring Static PW (Optional)

When you can access the Martini VPWS via CLI configuration, you can specify the mode as the static mode, and then configure the outgoing label and incoming label of the VC manually instead of adding the FTN and ILM options of the VC through Label Distribution Protocol (LDP).

Run the following commands to configure a VPWS static PW:

Command	Function
Ruijie# **config terminal**	Enters the global configuration mode.
Ruijie(config)# **interface vlan** *vlan-id*	Configures and enters the VLAN interface configuration mode.
Ruijie(config-if-*type ID*)# **xconnect** *vc_peer vc_id* **encapsulation mpls ethernet raw manual**	Creates a VC and configures the raw encapsulation mode.
Ruijie(config-if-*type ID*)# **exit**	Exits the interface configuration mode.
Ruijie(config)# **mpls static-l2vc** *ipv4-address vc-id* **out-label** *out-label-value* **in-label** *in-label-value*	Configures the outgoing label and incoming label of the static PW.
Ruijie(config-vfi)# **end**	Returns to the privileged EXEC mode.
Ruijie# **copy running-config startup-config**	Saves the configuration.

Caution

The static PW does not apply to heterogeneous media interconnection.
The outgoing and incoming labels of the static PW do not take effect on the VC instances of the Kompella VPWS. They only apply to the VC instances of the Martini VPWS, that is, the static forwarding entry instead of LDP signaling is adopted.

Maintaining PW Connectivity of VPWS

Configuration Prerequisites

When using VCCV to check the PW connectivity of VPWS, you must ensure that the VPWS network is already correctly configured.

Checking PW Connectivity

Command	Function
Ruijie# **ping mpls pseudowire** { *ip-address vc-id* [**segment** *segment-number*] | **kompella** *vfi_name local_site_id remote_site_id* **rd** *remote_rd* } { **label-alert** | **ttl-expiry** } [**repeat** *repeat*] [**timeout** *timeout*] [**size** *size*] [**interval** *mseconds*] [**source** *ip-address*] [**destination** *ip-address*] [**pad** *pattern*] [**reply mode** { **ipv4** | **router-alert | control-channel** }] [**verbose**]	Checks the PW connectivity.

Example: Check the connectivity of the PW whose peer PE IPv4 address is 10.10.10.3 and PW ID is 100.

```
Ruijie#ping mpls pseudowire 10.10.10.3 100 label-alert

Sending 5, 112-byte MPLS Echoes on Pseudowire, peer address 10.10.10.3, VC ID 100,

    timeout is 2 seconds, send interval is 0 msec:

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Press Ctrl+C to break.
!!!!!
Success rate is 100 percent(5/5),round-trip min/avg/max=20/36/60 ms
```

Locating Data Plane Fault of PW

Command	Function
Ruijie# **traceroute mpls pseudowire** { *ip-address vc-id* {**segment** *segment-number*} | **kompella** *vfi_name local_site_id remote_site_id* **rd** *remote_rd* } { **label-alert** | **ttl-expiry** } [**timeout** *timeout*] [**source** *ip-address*] [**destination** *ip-address*] [**reply mode** { **ipv4 | router-alert | control-channel** }] [**verbose**]	Traces the PW path.

Example: Track the path of the PW whose VFI name is vpn-1, local VE ID is 1, remote VE ID is 2, and remote RD is 200:1 to locate the fault.

```
Ruijie#traceroute mpls pseudowire kompella vpn-1 1 2 label-alert rd 200:1
```

```
Traceing Pseudowire Label Switched Path, VFI name vpn-1, local site id 1, remote site id 2,
timeout is 2 seconds.
Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Press Ctrl+C to break.
  0 192.168.1.10    MRU 1500 [Labels: router-alert/1538 Exp: 0/0]
! 1 192.168.1.11    20 ms
```

Verifying VPWS Configuration

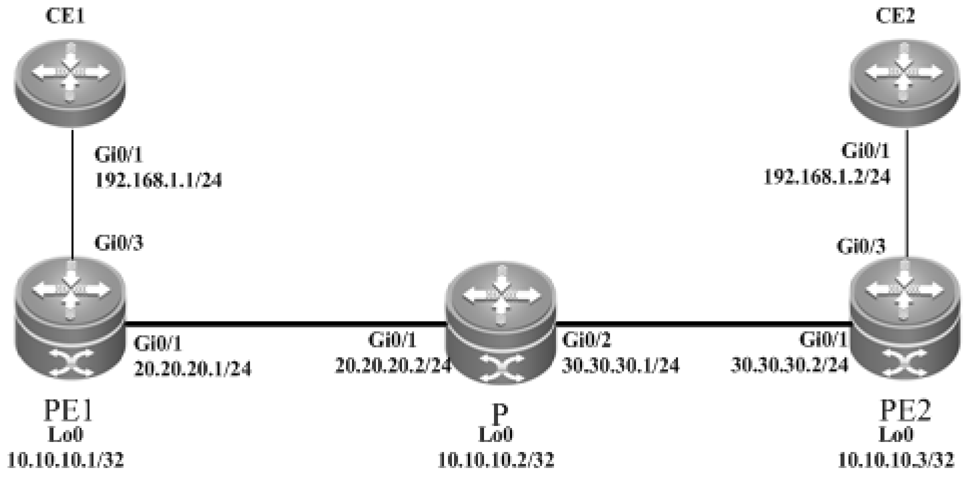
Command	Function
show bgp l2vpn vpws all	Displays NLRI information about all Kompella VPWS instances.
show bgp l2vpn vpws all connections	Displays Kompella VPWS signaling information.
show mpls l2transport vc [*vc_id* [*ip-address*]] | [**interface** *interface_name*] [**detail**]	Displays VC information established by VPWS.
show mpls vfi [*name*]	Displays VFI instance information of Kompella VPWS.
show mpls ldp vc	Displays VC signaling information of Martini VPWS.

Martini VPWS Router Configuration Instance

Connecting CEs to PEs Through Ethernet

As shown in the following network topology, the interface between PEs and CEs is an Ethernet interface. That is, CEs are connected to PEs through the Ethernet interface, through which L2VPN service is provided between CE1 and CE2.

Figure 53



The configuration procedure is as follows:

- Configuring CE1:

# Configure the interface that connects CE1 and PE1.

```
Ruijie(config)# interface gigabitethernet 0/1
Ruijie(config-if-Gigabitethernet 0/1)# ip address 192.168.1.1 255.255.255.0
```

# Enable the fast forwarding function of routers on the interface for routers. You do not need to use this command on switches.

```
Ruijie(config-if-gigabitethernet 0/1)# ip ref
Ruijie(config-if-Gigabitethernet 0/1)# end
```

- Configuring PE1:

# Configure the loopback interface.

```
Ruijie# configure terminal
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 10.10.10.1 255.255.255.255
```

# Configure the public network LSP tunnel and remote LDP neighbor.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# neighbor 10.10.10.3
Ruijie(config-mpls-router)# exit
Ruijie(config)# interface gigabitethernet 0/1
Ruijie(config-if-Gigabitethernet 0/1)# ip address 20.20.20.1 255.255.255.0
Ruijie(config-if-Gigabitethernet 0/1)# mpls ip
```

# Enable the fast forwarding function of routers on the interface for routers. You do not need to use this command on switches.

```
Ruijie(config-if-gigabitethernet 0/1)# ip ref
Ruijie(config-if-Gigabitethernet 0/1)# exit
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 20.20.20.0 0.0.0.255 area 0
Ruijie(config-router)# network 10.10.10.1 0.0.0.0 area 0
Ruijie(config-router)# end
```

# Configure the interface that connects PEs and CEs to enable the VPWS service.

```
Ruijie# configure terminal
Ruijie(config)# interface gigabitethernet 0/3
Ruijie(config-if-Gigabitethernet 0/3)# xconnect 10.10.10.3 2 encapsulation mpls ethernet raw
Ruijie(config-if-Gigabitethernet 0/3)# exit
```

- Configuring P:

# Configure the public network route protocol and LSP tunnel.

The configuration procedure is similar to the aforementioned MPLS basic function configuration instance.

- Configuring PE2:

The configuration is similar to that of PE1.

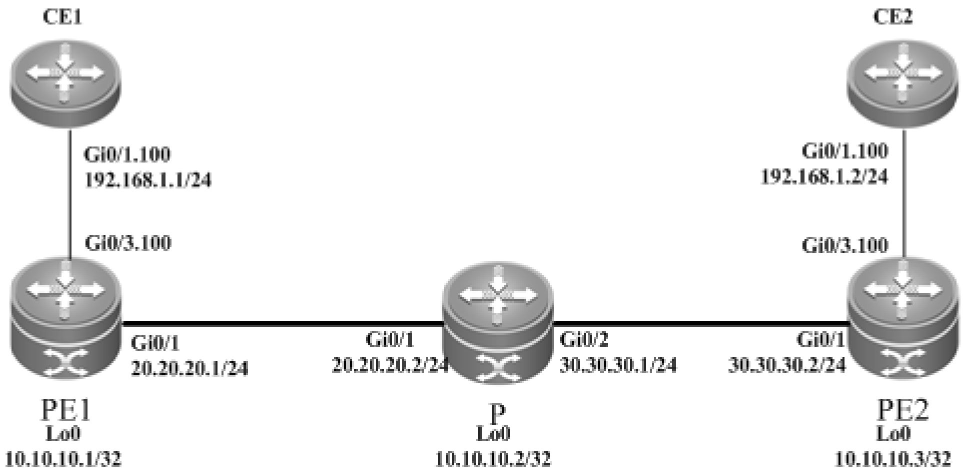
- Configuring CE2:

The configuration is similar to that of CE1.

Connecting CEs to PEs Through VLAN

As shown in the following network topology, the interface between PEs and CEs is an Ethernet sub-interface. That is, CEs are connected to PEs through the Ethernet sub-interface, through which L2VPN service is provided between CE1 and CE2.

Figure 54



The configuration procedure is as follows:

- Configuring CE1:

# Configure the interface that connects CE1 and PE2.

```
Ruijie(config)# interface gigabitethernet 0/1
```

# Enable the fast forwarding function of routers on the interface for routers. You do not need to use this command on switches.

```
Ruijie(config-if-gigabitethernet 0/1)# ip ref
Ruijie(config-if-gigabitethernet 0/1)# exit
Ruijie(config)# interface gigabitethernet 0/1.100
Ruijie(config-if-Gigabitethernet 0/1.100)# encapsulation dot1Q 100
Ruijie(config-if-Gigabitethernet 0/1.100)# ip address 192.168.1.1 255.255.255.0
Ruijie(config-if-Gigabitethernet 0/1.100)# end
```

- Configuring PE1:

# Configure the loopback interface.

```
Ruijie# configure terminal
```

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 10.10.10.1 255.255.255.255
```

# Configure the public network LSP tunnel and remote LDP neighbor.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# neighbor 10.10.10.3
Ruijie(config-mpls-router)# exit
Ruijie(config)# interface gigabitethernet 0/1
Ruijie(config-if-Gigabitethernet 0/1)# ip address 20.20.20.1 255.255.255.0
Ruijie(config-if-Gigabitethernet 0/1)# mpls ip
Ruijie(config-if-Gigabitethernet 0/1)# label-switching
```

# Enable the fast forwarding function of routers on the interface for routers. You do not need to use this command on switches.

```
Ruijie(config-if-gigabitethernet 0/1)# ip ref
Ruijie(config-if-Gigabitethernet 0/1)# exit
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 20.20.20.0 0.0.0.255 area 0
Ruijie(config-router)# network 10.10.10.1 0.0.0.0 area 0
Ruijie(config-router)# end
```

# Configure the interface that connects PEs and CEs to enable the VPWS service.

```
Ruijie# configure terminal
Ruijie(config)# interface gigabitethernet 0/3
Ruijie(config-if-Gigabitethernet 0/3)# ip ref
Ruijie(config-if-Gigabitethernet 0/3)# exit
Ruijie(config)# interface gigabitethernet 0/3.100
Ruijie(config-if-Gigabitethernet 0/3.100)# encapsulation dot1Q 100
Ruijie(config-if-Gigabitethernet 0/3.100)# xconnect 10.10.10.3 2 encapsulation mpls
ethernetvlan tagged
Ruijie(config-if-Gigabitethernet 0/3.100)# exit
```

- Configuring P:

# Configure the public network route protocol and LSP tunnel.

The configuration procedure is similar to the aforementioned MPLS basic function configuration instance.

- Configuring PE2:

The configuration is similar to that of PE1.

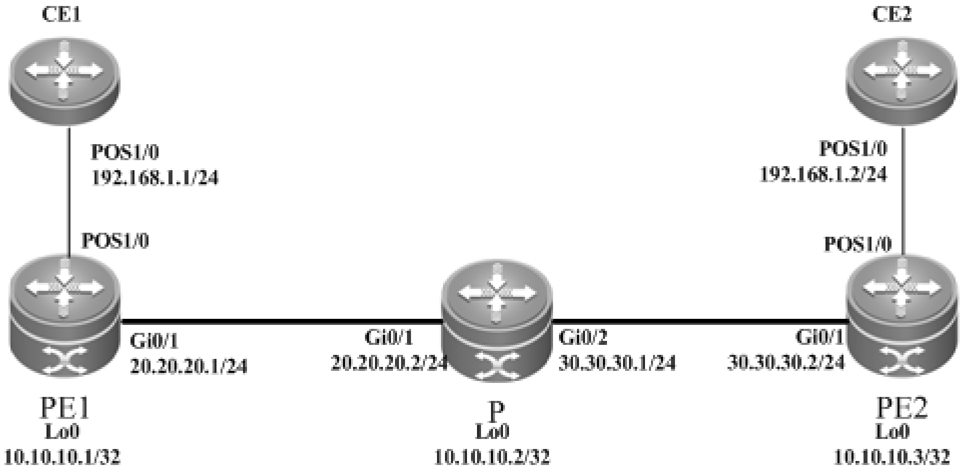
- Configuring CE2:

The configuration is similar to that of CE1.

Connecting CEs to PEs Through PPP

As shown in the following network topology, the interface between PEs and CEs is a POS interface. That is, the PPP link protocol is encapsulated and CEs are connected to PEs through PPP, through which L2VPN service is provided between CE1 and CE2.

Figure 55



The configuration procedure is as follows:

- Configuring CE1:

# Configure the interface that connects CE1 and PE2.

```
Ruijie(config)# interface pos 1/0
Ruijie(config-if-pos 1/0)# encapsulation ppp
Ruijie(config-if-pos 1/0)# ip address 192.168.1.1 255.255.255.0
Ruijie(config-if-pos 1/0)# clock internal
```

# Enable the fast forwarding function of routers on the interface for routers. You do not need to use this command on switches.

```
Ruijie(config-if-pos 1/0)# ip ref
Ruijie(config-if-pos 1/0)# end
```

- Configuring PE1:

# Configure the loopback interface.

```
Ruijie# configure terminal
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 10.10.10.1 255.255.255.255
```

# Configure the public network LSP tunnel and remote LDP neighbor.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# neighbor 10.10.10.3
Ruijie(config-mpls-router)# exit
Ruijie(config)# interface gigabitethernet 0/1
Ruijie(config-if-Gigabitethernet 0/1)# ip address 20.20.20.1 255.255.255.0
Ruijie(config-if-Gigabitethernet 0/1)# mpls ip
Ruijie(config-if-Gigabitethernet 0/1)# label-switching
```

# Enable the fast forwarding function of routers on the interface for routers. You do not need to use this command on switches.

```
Ruijie(config-if-gigabitethernet 0/1)# ip ref
Ruijie(config-if-Gigabitethernet 0/1)# exit
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 20.20.20.0 0.0.0.255 area 0
Ruijie(config-router)# network 10.10.10.1 0.0.0.0 area 0
Ruijie(config-router)# end
```

# Configure the interface that connects PEs and CEs to enable the VPWS service.

```
Ruijie(config)# interface pos 1/0
Ruijie(config-if-pos 1/0)# encapsulation ppp
Ruijie(config-if-pos 1/0)# clock internal
Ruijie(config-if-pos 1/0)# xconnect 10.10.10.3 2 encapsulation mpls ppp
Ruijie(config-if-pos 1/0)# end
```

- Configuring P:

# Configure the public network route protocol and LSP tunnel.

The configuration procedure is similar to the aforementioned MPLS basic function configuration instance.

- Configuring PE2:

The configuration is similar to that of PE1.

- Configuring CE2:

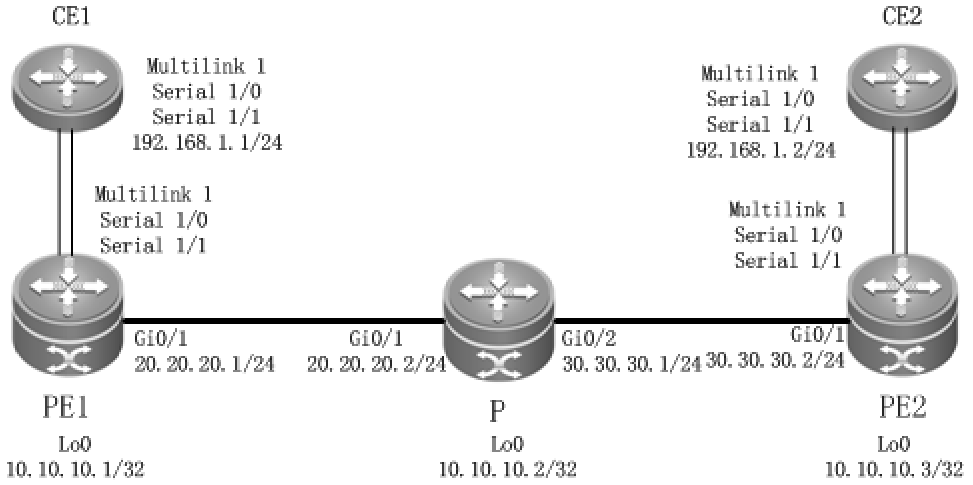
The configuration is similar to that of CE1.

Connecting CEs to PEs Through MultiPPP

As shown in the following network topology, PEs and CEs are connected through the Serial interface and the PPP protocol is encapsulated. PEs and CEs bind two Serial interfaces, which work in MultiPPP mode. Therefore, CEs are connected to PEs through MultiPPP to establish the L2VPN service between CE1 and CE2.

CEs are connected to PEs through MultiPPP by enabling the heterogeneous media access mode on the PE-end multilink interface. The mode does not support the homogeneous media access mode.

Figure 56



The configuration procedure is as follows:

- Configuring CE1:

# Configure the multilink interface that connects CE1 and PE2.

```
Ruijie(config)# interface mutilink 1
Ruijie(config-multilink 1)# ip address 192.168.1.1 255.255.255.0
```

# Enable the fast forwarding function of routers on the interface for routers. You do not use this command on switches.

```
Ruijie(config-multilink 1)# ip ref
Ruijie(config-multilink 1)# end
```

# Configure the serial interface bound with the multilink interface.

```
Ruijie(config)# interface serial 1/0
Ruijie(config-serial 1/0)# encapsulation ppp
Ruijie(config-serial 1/0)# ppp multilink
Ruijie(config-serial 1/0)# ppp multilink group 1
Ruijie(config-serial 1/0)# end
Ruijie(config)# interface serial 1/1
Ruijie(config-serial 1/1)# encapsulation ppp
Ruijie(config-serial 1/1)# ppp multilink
Ruijie(config-serial 1/1)# ppp multilink group 1
Ruijie(config-serial 1/1)# end
```

- Configuring PE1:

# Configure the loopback interface.

```
Ruijie# configure terminal
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 10.10.10.1 255.255.255.255
```

# Configure the public network LSP tunnel and remote LDP neighbor.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# neighbor 10.10.10.3
Ruijie(config-mpls-router)# exit
Ruijie(config)# interface gigabitethernet 0/1
Ruijie(config-if-Gigabitethernet 0/1)# ip address 20.20.20.1 255.255.255.0
Ruijie(config-if-Gigabitethernet 0/1)# mpls ip
Ruijie(config-if-Gigabitethernet 0/1)# label-switching
```

# Enable the fast forwarding function of routers on the interface for routers. You do not use this command on switches.

```
Ruijie(config-if-gigabitethernet 0/1)# ip ref
Ruijie(config-if-Gigabitethernet 0/1)# exit
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 20.20.20.0 0.0.0.255 area 0
Ruijie(config-router)# network 10.10.10.1 0.0.0.0 area 0
Ruijie(config-router)# end
```

# Configure the interface that connects PEs and CEs to enable the VPWS service.

# Configure the multilink interface.

```
Ruijie(config)# interface mutilink 1
Ruijie(config-multilink 1)# ip ref
Ruijie(config-multilink 1)# xconnect 10.10.10.3 2 encapsulation mpls ip-interworking
Ruijie(config-multilink 1)# ppp ipcp address proxy 192.168.1.2
```

# Configure the serial interface bound with the multilink interface.

```
Ruijie(config)# interface serial 1/0
Ruijie(config-serial 1/0)# encapsulation ppp
Ruijie(config-serial 1/0)# ppp multilink
Ruijie(config-serial 1/0)# ppp multilink group 1
Ruijie(config-serial 1/0)# end
Ruijie(config)# interface serial 1/1
Ruijie(config-serial 1/1)# encapsulation ppp
Ruijie(config-serial 1/1)# ppp multilink
Ruijie(config-serial 1/1)# ppp multilink group 1
Ruijie(config-serial 1/1)# end
```

- Configuring P:

# Configure the public network route protocol and LSP tunnel.

The configuration procedure is similar to the aforementioned MPLS basic function configuration instance.

- Configuring PE2:

The configuration procedure is similar to that of PE1.

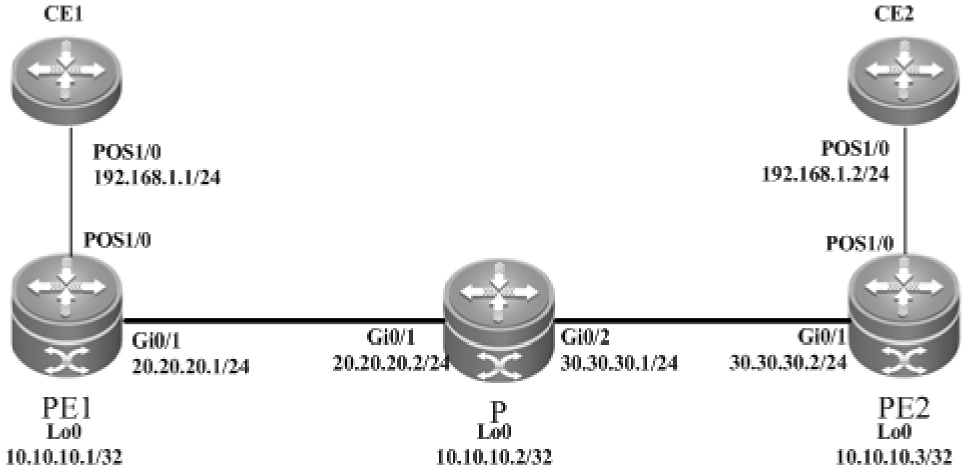
- Configuring CE2:

The configuration procedure is similar to that of CE1.

Connecting CEs to PEs Through HDLC

As shown in the following network topology, the interface that connects PEs and CEs is a POS interface and the HDLC link protocol is encapsulated. That is, CEs are connected to PEs through a HDLC interface, through which L2VPN service is provided between CE1 and CE2.

Figure 57



The configuration procedure is as follows:

- Configuring CE1:

# Configure the interface that connects CE1 and PE2.

```
Ruijie(config)# interface pos 1/0
Ruijie(config-if-pos 1/0)# encapsulation hdlc
Ruijie(config-if-pos 1/0)# ip address 192.168.1.1 255.255.255.0
Ruijie(config-if-pos 1/0)# clock internal
```

# Enable the fast forwarding function of routers on the interface for routers. You do not need to use this command on switches.

```
Ruijie(config-if-pos 1/0)# ip ref
Ruijie(config-if-pos 1/0)# end
```

- Configuring PE1:

# Configure the loopback interface.

```
Ruijie# configure terminal
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 10.10.10.1 255.255.255.255
```

# Configure the public network LSP tunnel and remote LDP neighbor.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# neighbor 10.10.10.3
Ruijie(config-mpls-router)# exit
Ruijie(config)# interface gigabitethernet 0/1
Ruijie(config-if-Gigabitethernet 0/1)# ip address 20.20.20.1 255.255.255.0
Ruijie(config-if-Gigabitethernet 0/1)# mpls ip
Ruijie(config-if-Gigabitethernet 0/1)# label-switching
```

# Enable the fast forwarding function of routers on the interface for routers. You do not need to use this command on switches.

```
Ruijie(config-if-gigabitethernet 0/1)# ip ref
Ruijie(config-if-Gigabitethernet 0/1)# exit
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 20.20.20.0 0.0.0.255 area 0
Ruijie(config-router)# network 10.10.10.1 0.0.0.0 area 0
Ruijie(config-router)# end
```

# Configure the interface that connects PEs and CEs to enable the VPWS service.

```
Ruijie(config)# interface pos 1/0
Ruijie(config-if-pos 1/0)# encapsulation hdlc
Ruijie(config-if-pos 1/0)# clock internal
Ruijie(config-if-pos 1/0)# xconnect 10.10.10.3 2 encapsulation mpls hdlc
Ruijie(config-if-pos 1/0)# end
```

- Configuring P:

# Configure the public network route protocol and LSP tunnel.

The configuration procedure is similar to the aforementioned MPLS basic function configuration instance.

- Configuring PE2:

The configuration is similar to that of PE1.

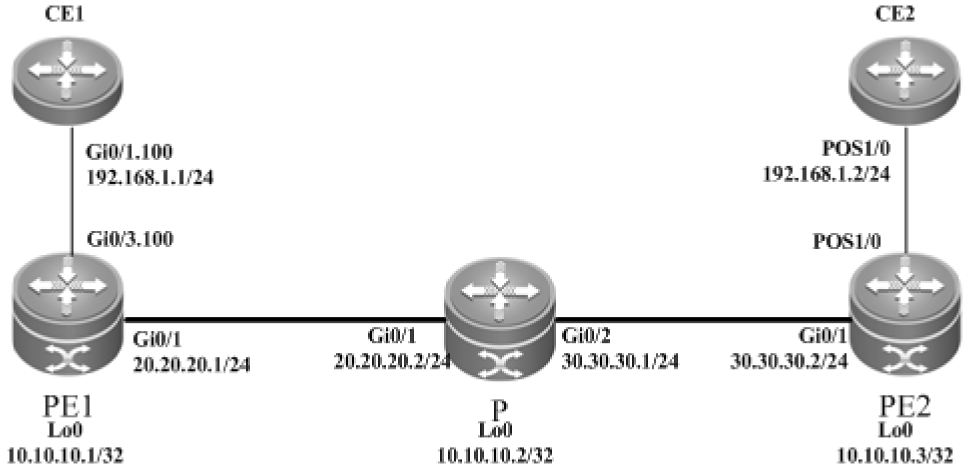
- Configuring CE2:

The configuration is similar to that of CE1.

VLAN and PPP Heterogeneous Media Communication

As shown in the following network topology, the interface that connects PE1 and CE1 is an Ethernet sub-interface and the encapsulated link protocol is 802.1Q. PE2 and CE2 are connected through the POS interface and the encapsulated link protocol is PPP, through which the L2VPN service is established between CE1 and CE2. VPWS provides the heterogeneous media communication L2VPN service.

Figure 58



The configuration procedure is as follows:

# Configure the interface that connects CE1 and PE1.

```
Ruijie(config)# interface gigabitethernet 0/1
```

# Enable the fast forwarding function of routers on the interface for routers. You do not need to use this command on switches.

```
Ruijie(config-if-gigabitethernet 0/1)# ip ref
Ruijie(config-if-gigabitethernet 0/1)# exit
Ruijie(config)# interface gigabitethernet 0/1.100
Ruijie(config-if-Gigabitethernet 0/1.100)# encapsulation dot1Q 100
Ruijie(config-if-Gigabitethernet 0/1.100)# ip address 192.168.1.1 255.255.255.0
Ruijie(config-if-Gigabitethernet 0/1.100)# end
```

- Configuring PE1:

# Configure the loopback interface.

```
Ruijie# configure terminal
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 10.10.10.1 255.255.255.255
```

# Configure the public network LSP tunnel and remote LDP neighbor.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# neighbor 10.10.10.3
Ruijie(config-mpls-router)# exit
Ruijie(config)# interface gigabitethernet 0/1
Ruijie(config-if-Gigabitethernet 0/1)# ip address 20.20.20.1 255.255.255.0
Ruijie(config-if-Gigabitethernet 0/1)# mpls ip
Ruijie(config-if-Gigabitethernet 0/1)# label-switching
```

# Enable the fast forwarding function of routers on the interface for routers. You do not need to use this command on switches.

```
Ruijie(config-if-gigabitethernet 0/1)# ip ref
Ruijie(config-if-Gigabitethernet 0/1)# exit
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 20.20.20.0 0.0.0.255 area 0
Ruijie(config-router)# network 10.10.10.1 0.0.0.0 area 0
Ruijie(config-router)# end
```

# Configure the interface that connects PEs and CEs to enable the VPWS service.

```
Ruijie# configure terminal
Ruijie(config)# interface gigabitethernet 0/3
Ruijie(config-if-Gigabitethernet 0/3)# ip ref
Ruijie(config-if-Gigabitethernet 0/3)# exit
Ruijie(config)# interface gigabitethernet 0/3.100
Ruijie(config-if-Gigabitethernet 0/3.100)# encapsulation dot1Q 100
Ruijie(config-if-Gigabitethernet 0/3.100)# xconnect 10.10.10.3 2 encapsulation mpls
ip-interworking local-ce mac 00d0.f811.2111
Ruijie(config-if-Gigabitethernet 0/3.100)# exit
```

- Configuring P:

# Configure the public network route protocol and LSP tunnel.

The configuration procedures is similar to the aforementioned MPLS basic function configuration instance.

- Configuring PE2:

# Configure the loopback interface.

```
Ruijie# configure terminal
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 10.10.10.3 255.255.255.255
```

# Configure the public network LSP tunnel and remote LDP neighbor.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# neighbor 10.10.10.1
Ruijie(config-mpls-router)# exit
Ruijie(config)# interface gigabitethernet 0/1
Ruijie(config-if-Gigabitethernet 0/1)# ip address 30.30.30.2 255.255.255.0
Ruijie(config-if-Gigabitethernet 0/1)# mpls ip
Ruijie(config-if-Gigabitethernet 0/1)# label-switching
```

# Enable the fast forwarding function of routers on the interface for routers. You do not need to use this command on switches.

```
Ruijie(config-if-gigabitethernet 0/1)# ip ref
Ruijie(config-if-Gigabitethernet 0/1)# exit
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 30.30.30.0 0.0.0.255 area 0
Ruijie(config-router)# network 10.10.10.3 0.0.0.0 area 0
Ruijie(config-router)# end
```

# Configure the interface that connects PEs and CEs to enable the VPWS service.

```
Ruijie# configure terminal
Ruijie(config)# interface pos 1/0
Ruijie(config-if-pos1/0)# ip ref
Ruijie(config-if-pos1/0)# encapsulation ppp
Ruijie(config-if-pos1/0)# ppp ipcp address proxy 192.168.1.1
Ruijie(config-if- pos1/0)# xconnect 10.10.10.3 2 encapsulation mpls ip-interworking
Ruijie(config-if-pos1/0)# exit
```

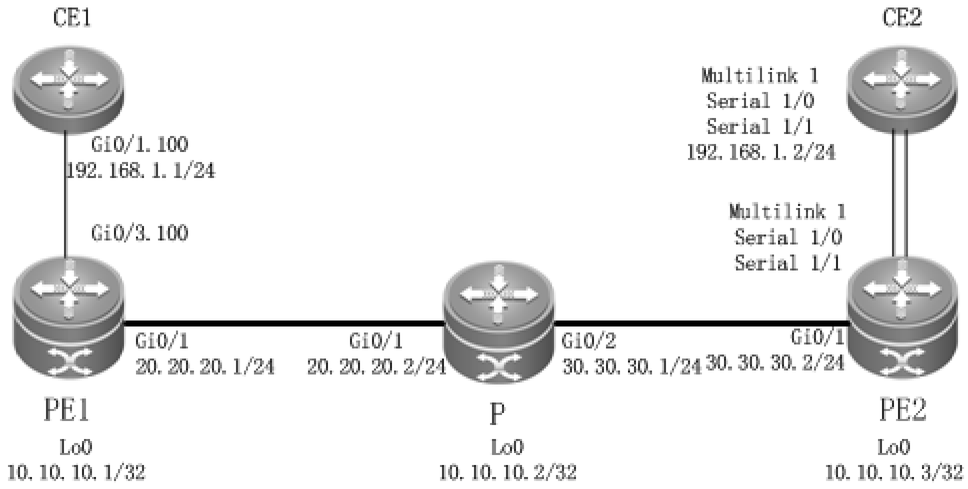
- Configuring CE2:

```
Ruijie(config)# interface pos 1/0
Ruijie(config-if-pos 1/0)# ip ref
Ruijie(config-if-pos 1/0)# encapsulation ppp
Ruijie(config-if-pos 1/0)# ip address 192.168.1.2 255.255.255.0
Ruijie(config-if-pos 1/0)# clock internal
Ruijie(config-if-pos 1/0)# end
```

VLAN and MultiPPP Heterogeneous Media Communication

As shown in the following network topology, the interface that connects PE1 and CE1 is an Ethernet sub-interface and the encapsulated link protocol is 802.1Q. PE2 and CE2 are connected through the Serial interface and the encapsulated link protocol is PPP. PPP is bound by two physical lines in the MultiPPP working mode, through which the L2VPN service is established between CE1 and CE2. VPWS provides the heterogeneous media communication L2VPN service.

Figure 59



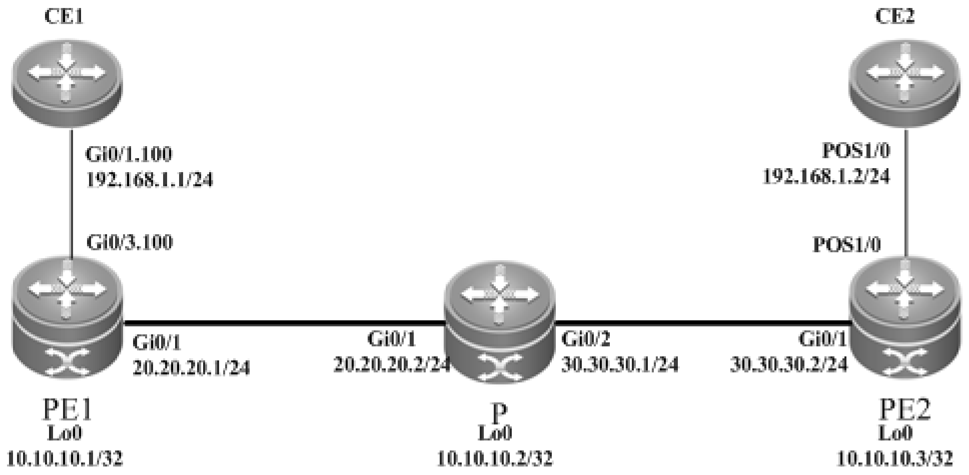
The configuration of CE1 and PE1 is generally the same as the configuration of VLAN and PPP heterogeneous media communication.

The configuration of CE2 and PE2 is generally the same as the configuration of a CE when it is connected to a PE through MultiPPP.

VLAN and HDLC Heterogeneous Media Communication

As shown in the following network topology, the interface between PE1 and CE1 is an Ethernet sub-interface and the encapsulated link protocol is 802.1Q. PE2 and CE2 are connected through the POS interface and the encapsulated link protocol is HDLC, through which the L2VPN service is established between CE1 and CE2. VPWS provides the heterogeneous media communication L2VPN service.

Figure 60



The configuration procedure is as follows:

# Configure the interface that connects CE1 and PE1.

```
Ruijie(config)# interface gigabitethernet 0/1
```

# Enable the fast forwarding function of routers on the interface for routers. You do not need to use this command on switches.

```
Ruijie(config-if-gigabitethernet 0/1)# ip ref
Ruijie(config-if-gigabitethernet 0/1)# exit
Ruijie(config)# interface gigabitethernet 0/1.100
Ruijie(config-if-Gigabitethernet 0/1.100)# encapsulation dot1Q 100
Ruijie(config-if-Gigabitethernet 0/1.100)# ip address 192.168.1.1 255.255.255.0
Ruijie(config-if-Gigabitethernet 0/1.100)# end
```

- Configuring PE1:

# Configure the loopback interface.

```
Ruijie# configure terminal
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 10.10.10.1 255.255.255.255
```

# Configure the public network LSP tunnel and remote LDP neighbor.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# neighbor 10.10.10.3
Ruijie(config-mpls-router)# exit
Ruijie(config)# interface gigabitethernet 0/1
Ruijie(config-if-Gigabitethernet 0/1)# ip address 20.20.20.1 255.255.255.0
Ruijie(config-if-Gigabitethernet 0/1)# mpls ip
Ruijie(config-if-Gigabitethernet 0/1)# label-switching
```

# Enable the fast forwarding function of routers on the interface for routers. You do not need to use this command on switches.

```
Ruijie(config-if-gigabitethernet 0/1)# ip ref
Ruijie(config-if-Gigabitethernet 0/1)# exit
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 20.20.20.0 0.0.0.255 area 0
Ruijie(config-router)# network 10.10.10.1 0.0.0.0 area 0
Ruijie(config-router)# end
```

# Configure the interface that connects PEs and CEs to enable the VPWS service.

```
Ruijie# configure terminal
Ruijie(config)# interface gigabitethernet 0/3
Ruijie(config-if-Gigabitethernet 0/3)# ip ref
Ruijie(config-if-Gigabitethernet 0/3)# exit
Ruijie(config)# interface gigabitethernet 0/3.100
Ruijie(config-if-Gigabitethernet 0/3.100)# encapsulation dot1Q 100
Ruijie(config-if-Gigabitethernet 0/3.100)# xconnect 10.10.10.3 2 encapsulation mpls
ip-interworking local-ce mac 00d0.f811.2111
Ruijie(config-if-Gigabitethernet 0/3.100)# exit
```

- Configuring P:

# Configure the public network route protocol and LSP tunnel.

The configuration procedure is similar to the aforementioned MPLS basic function configuration instance.

- Configuring PE2:

# Configure the loopback interface.

```
Ruijie# configure terminal
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 10.10.10.3 255.255.255.255
```

# Configure the public network LSP tunnel and remote LDP neighbor.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# neighbor 10.10.10.1
Ruijie(config-mpls-router)# exit
Ruijie(config)# interface gigabitethernet 0/1
Ruijie(config-if-Gigabitethernet 0/1)# ip address 30.30.30.2 255.255.255.0
Ruijie(config-if-Gigabitethernet 0/1)# mpls ip
Ruijie(config-if-Gigabitethernet 0/1)# label-switching
```

# Enable the fast forwarding function of routers on the interface for routers. You do not need to use this command on switches.

```
Ruijie(config-if-gigabitethernet 0/1)# ip ref
Ruijie(config-if-Gigabitethernet 0/1)# exit
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 30.30.30.0 0.0.0.255 area 0
Ruijie(config-router)# network 10.10.10.3 0.0.0.0 area 0
Ruijie(config-router)# end
```

# Configure the interface that connects PEs and CEs to enable the VPWS service.

```
Ruijie# configure terminal
Ruijie(config)# interface pos 1/0
Ruijie(config-if-pos 1/0)# ip ref
Ruijie(config-if-pos1/0)# encapsulation hdlc
Ruijie(config-if- pos1/0)# xconnect 10.10.10.3 2 encapsulation mpls ip-interworking
Ruijie(config-if-pos1/0)# exit
```

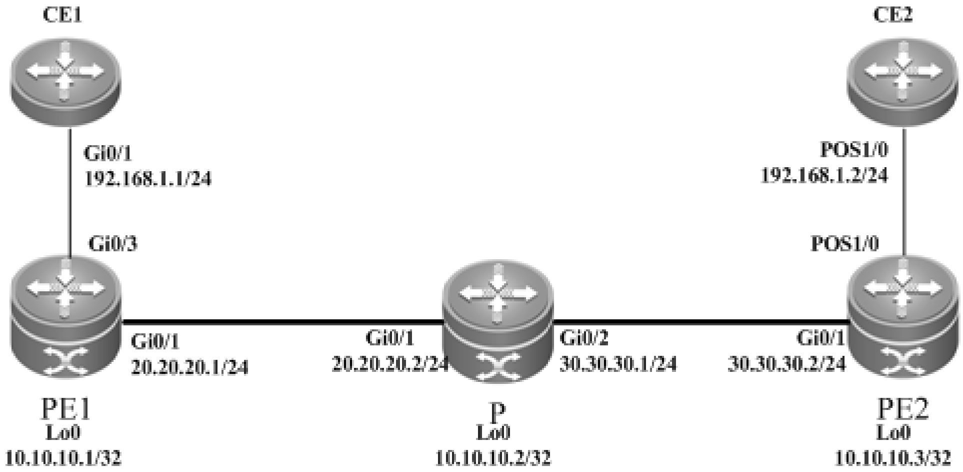
- Configuring CE2:

```
Ruijie(config)# interface pos 1/0
Ruijie(config-if-pos 1/0)# ip ref
Ruijie(config-if-pos 1/0)# encapsulation hdlc
Ruijie(config-if-pos 1/0)# ip address 192.168.1.2 255.255.255.0
Ruijie(config-if-pos 1/0)# clock internal
Ruijie(config-if-pos 1/0)# end
```

Ethernet and PPP Heterogeneous Media Communication

As shown in the following network topology, the interface between PE1 and CE1 is an Ethernet interface and the encapsulated link protocol is EthernetII. PE2 and CE2 are connected through the POS interface and the encapsulated link protocol is PPP, through which the L2VPN service is established between CE1 and CE2. VPWS provides the heterogeneous media communication L2VPN service.

Figure 61

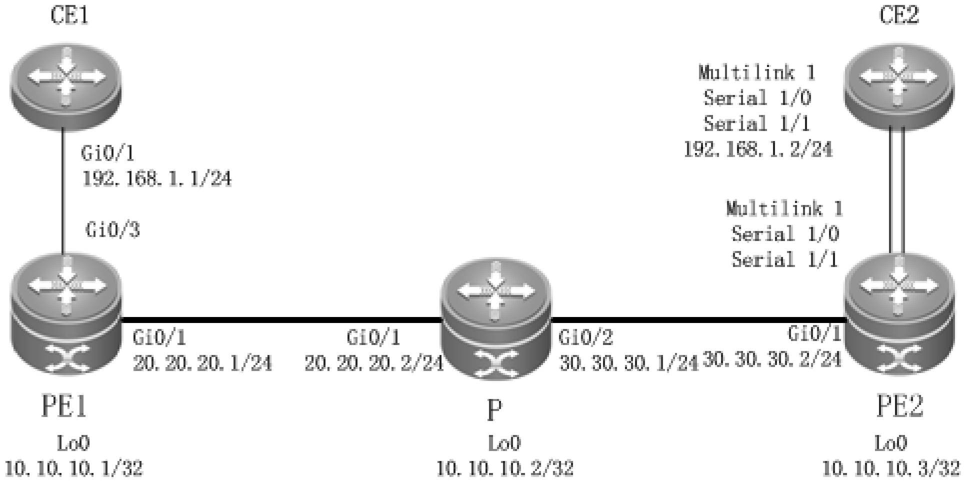


The configuration is generally the same as that for VLAN and PPP heterogeneous media communication, except the access modes of PE1 and CE1.

Ethernet and MultiPPP Heterogeneous Media Communication

As shown in the following network topology, the interface between PE1 and CE1 is an Ethernet interface and the encapsulated link protocol is EthernetII. PE2 and CE2 are connected through the Serial interface and the encapsulated link protocol is PPP. PPP is bound by two physical lines in the MultiPPP working mode, through which the L2VPN service is established between CE1 and CE2. VPWS provides the heterogeneous media communication L2VPN service.

Figure 62



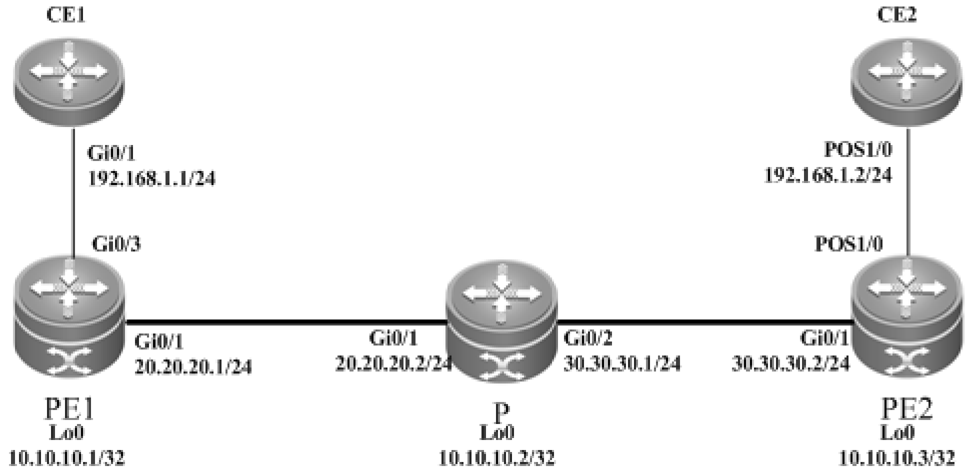
The configuration of CE1 and PE1 is generally the same as that for VLAN and PPP heterogeneous media communication, except the access mode of PE1 and CE1.

The configuration of CE2 and PE2 is generally the same as the configuration of a CE when it is connected to a PE through MultiPPP.

Ethernet and HDLC Heterogeneous Media Communication

As shown in the following network topology, the interface between PE1 and CE1 is an Ethernet interface and the encapsulated link protocol is EthernetII. PE2 and CE2 are connected through the POS interface and the encapsulated link protocol is HDLC, through which the L2VPN service is established between CE1 and CE2. VPWS provides the heterogeneous media communication L2VPN service.

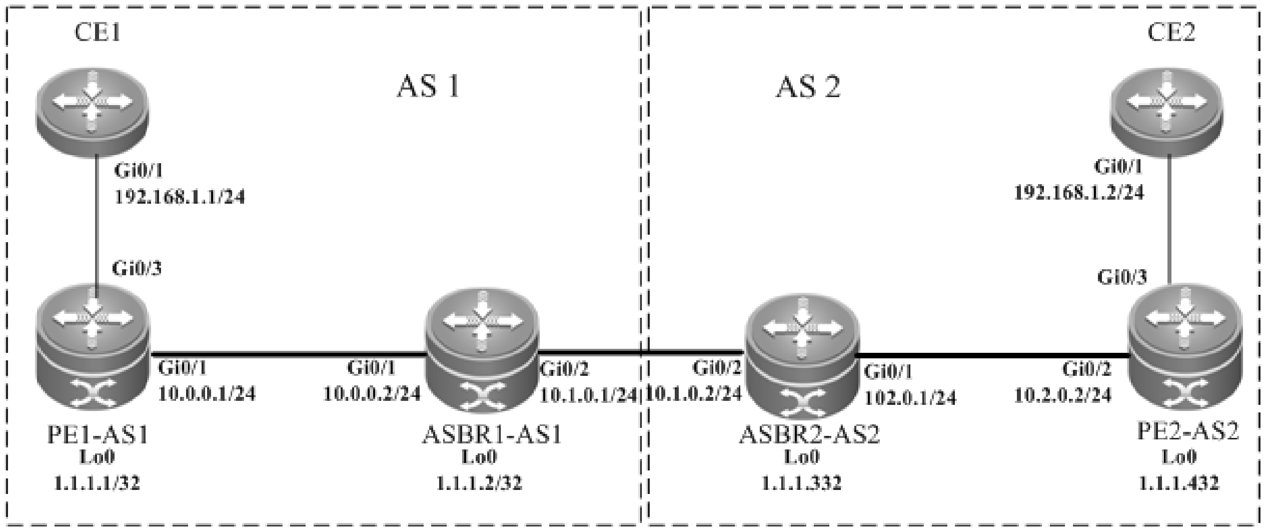
Figure 63



The configuration is generally the same as the configuration of VLAN and HDLC heterogeneous media communication, except the access modes of PE1 and CE1.

Option C Inter-AS VPWS

Figure 64 Option C: Inter-AS VPWS



The proceeding figure shows how to realize inter-AS VPWS by using the Option C solution. To set up a PW between PEs of different ASs, PW information is not maintained on ASBR and OSPF is used in each AS as IGP to realize inter-AS communication. Assuming that CE1 is connected to a PE through the master port, it is required to establish L2VPN communication between CE1 and CE2. Configuration of devices is described as follows (only configuration related to the function is included):

- Configuring CE1:

```
Ruijie# configure terminal
Ruijie(config)# interface GigabitEthernet 0/1
```

# The **no switchport** command is used on switches to switch the port mode to the Routed Port mode. It does not apply to routers and does not need to be used on routers.

```
Ruijie(config-if-GigabitEthernet 0/1)# no switchport
Ruijie(config-if-GigabitEthernet 0/1)# ip address 192.168.1.1 255.255.255.0
```

# Enable the fast forwarding function of routers on the interface for routers. You do not need to use this command on switches.

```
Ruijie(config-if-gigabitethernet 0/1)# ip ref
Ruijie(config-if-GigabitEthernet 0/1)# end
```

The configuration of CE2 is similar to that of CE1.

- Configuring PE1-AS1:

# Configure the public network route protocol, MPLS signaling, and remote neighbors.

```
Ruijie# configure terminal
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 1.1.1.1 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# neighbor 1.1.1.4
Ruijie(config-mpls-router)# exit
Ruijie(config)# router ospf 1
Ruijie(config-router)# network 10.0.0.0 0.0.0.255 area 0
Ruijie(config-router)# network 1.1.1.1 0.0.0.0 area 0
Ruijie(config-router)# exit
Ruijie(config)# interface GigabitEthernet 0/1
```

# The **no switchport** command is used on switches to switch the port mode to the Routed Port mode. It does not apply to routers and does not need to be used on routers.

```
Ruijie(config-if-GigabitEthernet 0/1)# no switchport
```

# Enable the fast forwarding function of routers on the interface for routers. You do not need to use this command on switches.

```
Ruijie(config-if-gigabitethernet 0/1)# ip ref
Ruijie(config-if-GigabitEthernet 0/1)# ip address 10.0.0.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 0/1)# mpls ip
Ruijie(config-if-GigabitEthernet 0/1)# label-switching
Ruijie(config-if-GigabitEthernet 0/1)# end
```

# Configure BGP.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 1.1.1.2 remote-as 1
Ruijie(config-router)# neighbor 1.1.1.2 update-source loopback 0
Ruijie(config-router)# address-family ipv4
Ruijie(config-router-af)# neighbor 1.1.1.2  activate
Ruijie(config-router-af)# neighbor 1.1.1.2  send-label
Ruijie(config-router-af)# end
```

# Configure the user access VPWS.

```
Ruijie# configure terminal
Ruijie(config)# interface GigabitEthernet 0/3
```

# Enable the fast forwarding function of routers on the interface for routers. You do not need to use this command on switches.

```
Ruijie(config-if-gigabitethernet 0/3)# ip ref
Ruijie(config-if-GigabitEthernet 0/3)# xconnect 1.1.1.1 1 encapsulation mpls ethernet
Ruijie(config-if-GigabitEthernet 0/3)# end
```

The configuration of PE2-AS2 is similar to that of PE1-AS1.

- Configuring ASBR1-AS1:

# Configure the loopback interface.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 1.1.1.2 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure the public network route protocol and MPLS signaling.

```
Ruijie# configure terminal
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# exit
Ruijie(config)# interface GigabitEthernet 0/1
```

# The **no switchport** command is used on switches to switch the port mode to the Routed Port mode. It does not apply to routers and does not need to be used on routers.

```
Ruijie(config-if-GigabitEthernet 0/1)# no switchport
```

# Enable the fast forwarding function of routers on the interface for routers. You do not need to use this command on switches.

```
Ruijie(config-if-gigabitethernet 0/3)# ip ref
Ruijie(config-if-GigabitEthernet 0/1)# ip address 10.0.0.2 255.255.255.0
Ruijie(config-if-GigabitEthernet 0/1)# mpls ip
Ruijie(config-if-GigabitEthernet 0/1)# label-switching
Ruijie(config-if-GigabitEthernet 0/1)# exit
Ruijie(config)# router ospf 1
Ruijie(config-router)# network 10.0.0.0 0.0.0.255 area 0
Ruijie(config-router)# network 1.1.1.2 0.0.0.0 area 0
Ruijie(config-router)# exit
```

# Configure the IP address of the interface connected to ASBR2.

```
Ruijie(config)# interface GigabitEthernet 0/2
```

# The **no switchport** command is used on switches to switch the port mode to the Routed Port mode. It does not apply to routers and does not need to be used on routers.

```
Ruijie(config-if-GigabitEthernet 0/2)# no switchport
```

# Enable the fast forwarding function of routers on the interface for routers. You do not need to use this command on switches.

```
Ruijie(config-if-gigabitethernet 0/2)# ip ref
Ruijie(config-if-GigabitEthernet 0/2)# ip address 10.1.0.1 255.255.255.0
```

# Enable the interface's label packet forwarding capability.

```
Ruijie(config-if-GigabitEthernet 0/2)# label-switching
Ruijie(config-if-GigabitEthernet 0/2)# exit
```

# Configure BGP.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 1
Ruijie(config-router)# neighbor 1.1.1.1 remote-as 1
Ruijie(config-router)# neighbor 1.1.1.1 update-source loopback 0
Ruijie(config-router)# neighbor 10.1.0.2 remote-as 2
Ruijie(config-router)# address-family ipv4
Ruijie(config-router-af)# neighbor 1.1.1.1 activate
Ruijie(config-router-af)# neighbor 1.1.1.1 send-label
Ruijie(config-router-af)# neighbor 10.1.0.2 activate
Ruijie(config-router-af)# neighbor 10.1.0.2 send-label
Ruijie(config-router-af)# network 1.1.1.1 mask 255.255.255.255
Ruijie(config-router-af)# end
```

The configuration of ASBR2-AS2 is similar to that of ASBR1-AS1.

The configuration of PE2-AS2 is similar to that of PE1-AS1.

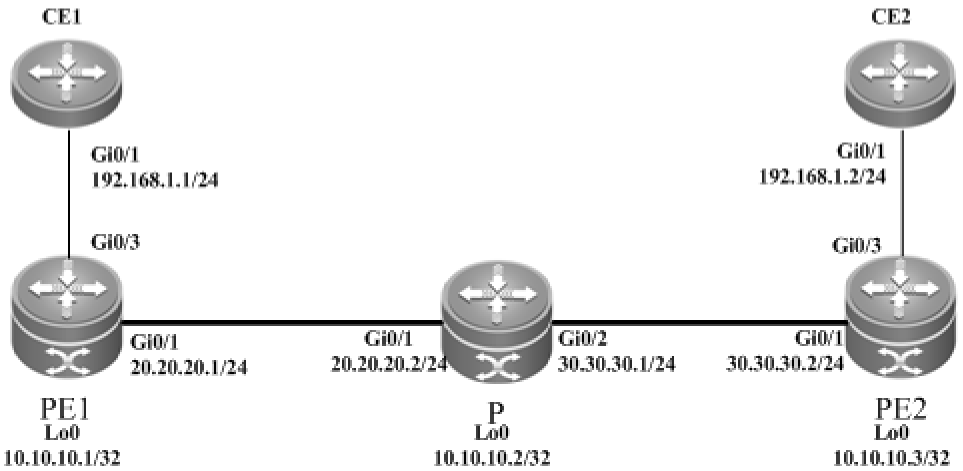
The configuration of CE2 is similar to that of CE1.

Kompella VPWS Router Configuration Instance

Connecting CEs to PEs Through Ethernet

As shown in the following network topology, the interface between PEs and CEs is an Ethernet interface. That is, CEs are connected to PEs through the Ethernet interface, through which the L2VPN service is provided between CE1 and CE2.

Figure 65



The configuration procedure is as follows:

Configuring CE1:

# Configure the interface between CE1 and PE1.

```
Ruijie(config)# interface gigabitethernet 0/1
Ruijie(config-if-Gigabitethernet 0/1)# ip address 192.168.1.1 255.255.255.0
```

# Enable the fast forwarding function of routers on the interface for routers. You do not need to use this command on switches.

```
Ruijie(config-if-gigabitethernet 0/1)# ip ref
Ruijie(config-if-Gigabitethernet 0/1)# end
```

Configuring PE1:

# Configure the loopback interface address.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 10.10.10.1 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 10.10.10.1 0.0.0.0 area 0
Ruijie(config-router)# network 20.20.20.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# exit
```

# Configure the public network tunnel between PEs.

```
Ruijie(config)# interface gigabitEthernet 0/1
Ruijie(config-if-Gigabitethernet 0/1)# ip ref
Ruijie(config-if-Gigabitethernet 0/1)# ip address 20.20.20.1 255.255.255.0
Ruijie(config-if-Gigabitethernet 0/1)# mpls ip
Ruijie(config-if-Gigabitethernet 0/1)# label-switching
Ruijie(config-if-Gigabitethernet 0/1)# exit
```

# Configure the L2VPN address family.

```
Ruijie(config)# router bgp 100
Ruijie(config-router)# neighbor 10.10.10.3 remote-as 100
Ruijie(config-router)# neighbor 10.10.10.3 update-source loopback 0
Ruiije(config-router)# address-family l2vpn vpws
Ruijie(config-router-af)# neighbor 10.10.10.3 activate
Ruijie(config-router-af)# neighbor 10.10.10.3 send-community extended
Ruijie(config-router-af)# exit
```

# Configure a VFI instance.

```
Ruijie# configure terminal
Ruijie(config)# l2 vfi vpws-1 vpnid 1 point-to-point
Ruijie(config-vfi)# rd 1:1
Ruijie(config-vfi)# encapsulation mpls ethernet
Ruijie(config-vfi)# route-target both 10000:1
Ruijie(config-vfi)# site-id 1
Ruijie(config-vfi-site)# xconnect interface gigabitEthernet 0/3 remote-ce-id 2
Ruijie(config-vfi-site)#exit-site-mode
Ruijie(config-vfi)#exit
```

# Enable the interface to connect to remote CEs.

```
Ruijie(config)# interface gigabitEthernet 0/3
Ruijie(config-if-GigabitEthernet 0/3)# ip ref
Ruijie(config-if-GigabitEthernet 0/3)# exit
```

Configuring P:

# Configure the public network route protocol and LSP tunnel.

The configuration is similar to the aforementioned MPLS basic function configuration instance.

Configuring PE2:

The configuration is similar to that of PE1.

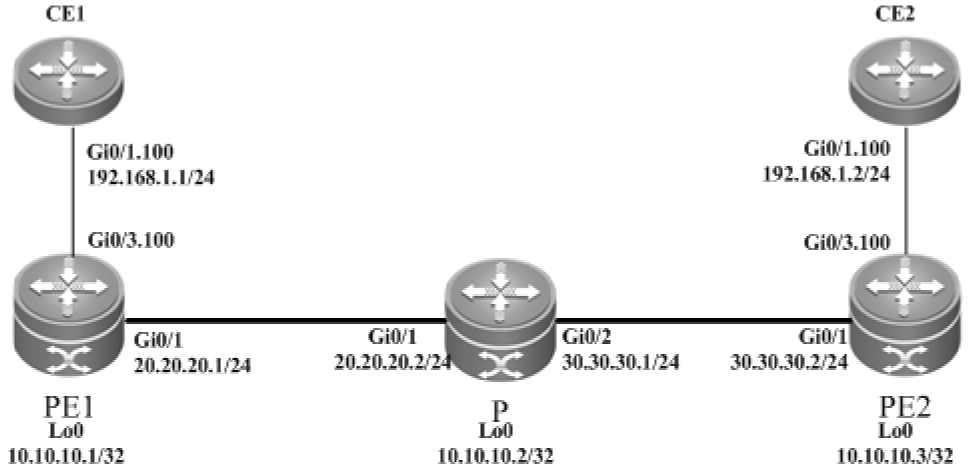
Configuring CE2:

The configuration is similar to that of CE1.

Connecting CEs to PEs Through VLAN

As shown in the following network topology, the interface that connects PEs and CEs is an Ethernet sub-interface. That is, CEs are connected to PEs through the Ethernet sub-interface, through which L2VPN service is provided between CE1 and CE2.

Figure 66



The configuration procedure is as follows:

Configuring CE1:

# Configure the interface between CE1 and PE1.

```
Ruijie(config)# interface gigabitethernet 0/1
```

# Enable the fast forwarding function of routers on the interface for routers. You do not need to use this command on switches.

```
Ruijie(config-if-gigabitethernet 0/1)# ip ref
Ruijie(config-if-gigabitethernet 0/1)# exit
Ruijie(config)# interface gigabitethernet 0/1.100
Ruijie(config-if-Gigabitethernet 0/1.100)# encapsulation dot1Q 100
Ruijie(config-if-Gigabitethernet 0/1.100)# ip address 192.168.1.1 255.255.255.0
Ruijie(config-if-Gigabitethernet 0/1.100)# end
```

Configuring PE1:

# Configure the loopback interface address.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 10.10.10.1 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 10.10.10.1 0.0.0.0 area 0
Ruijie(config-router)# network 20.20.20.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# exit
```

#Configure the public network tunnel between PEs.

```
Ruijie(config)# interface gigabitEthernet 0/1
Ruijie(config-if-Gigabitethernet 0/1)# ip ref
Ruijie(config-if-Gigabitethernet 0/1)# ip address 20.20.20.1 255.255.255.0
Ruijie(config-if-Gigabitethernet 0/1)# mpls ip
Ruijie(config-if-Gigabitethernet 0/1)# label-switching
Ruijie(config-if-Gigabitethernet 0/1)# exit
```

# Configure the L2VPN address family.

```
Ruijie(config)# router bgp 100
Ruijie(config-router)# neighbor 10.10.10.3 remote-as 100
Ruijie(config-router)# neighbor 10.10.10.3 update-source loopback 0
Ruiije(config-router)# address-family l2vpn vpws
Ruijie(config-router-af)# neighbor 10.10.10.3 activate
Ruijie(config-router-af)# neighbor 10.10.10.3 send-community extended
Ruijie(config-router-af)# exit
```

# Configure the interface to connect with remote CEs.

```
Ruijie(config-if-Gigabitethernet 0/3.100)# encapsulation dot1Q 100
Ruijie(config-if-GigabitEthernet 0/3)# exit
```

# Configure a VFI instance.

```
Ruijie# configure terminal
Ruijie(config)# l2 vfi vpws-1 vpnid 1 point-to-point
Ruijie(config-vfi)# rd 1:1
Ruijie(config-vfi)# encapsulation mpls ethernetvlan
Ruijie(config-vfi)# route-target both 10000:1
Ruijie(config-vfi)# site-id 1
Ruijie(config-vfi-site)# xconnect interface gigabitEthernet 0/3.100 remote-ce-id 2
Ruijie(config-vfi-site)#exit-site-mode
Ruijie(config-vfi)#exit
```

Configuring P:

# Configure the public network route protocol and LSP tunnel.

The configuration is similar to the aforementioned MPLS basic function configuration instance.

Configuring PE2:

The configuration procedure is similar to that of PE1.

Configuring CE2:

The configuration procedure is similar to that of CE1.

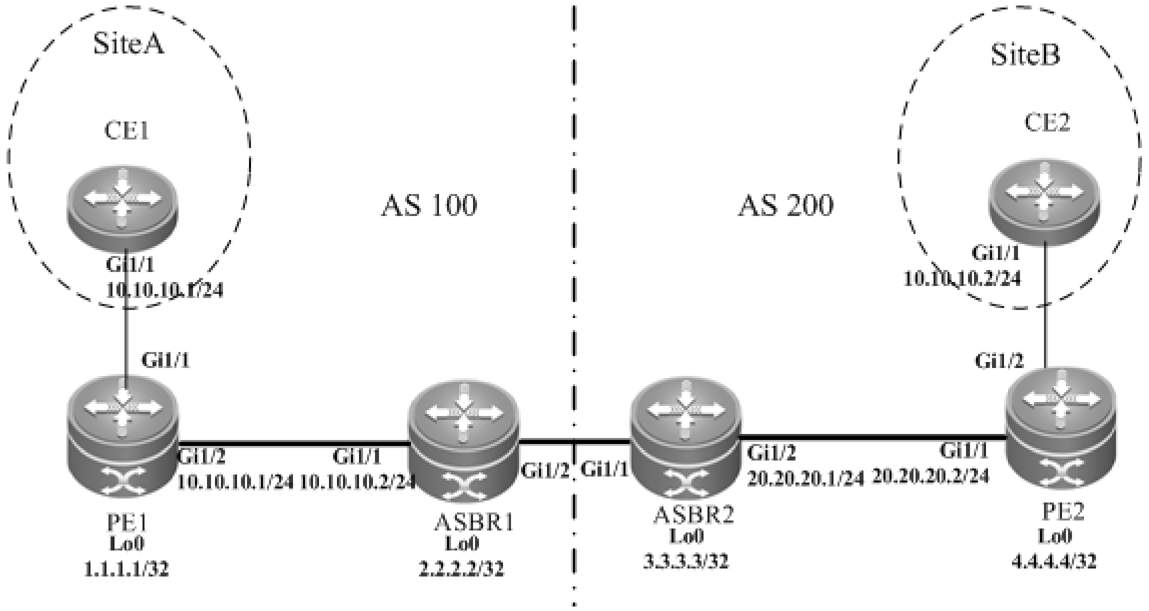
Option A Inter-AS VPWS

Networking Requirements

- CEs of customer S in site A and site B are connected with each other through the carrier's PE1 in AS 100 and PE2 in AS 200, realizing layer-2 point-to-point connection.
- PE1 and PE2 are in different autonomous ASs. ASBR1 and ASBR2 are considered CEs by each other, which means that the interface between ASBRs connects the AC to the VFI instance.

Networking Topology

Figure 67 Kompella VPWS inter-AS networking topology



The preceding figure shows the structure of the Kompella VPLS inter-AS networking topology in Option A. The intermediate interface is considered by ASBRs as AC access.

Configuration Tips

Before configuring Kompella VPWS, complete the following tasks:

- Run IGP in the carrier's network to realize connection between PE and ASBR devices.
- Establish the MP-IBGP peer relationship between PEs and intra-AS ASBRs.
- Obtain Kompella VPWS configuration information including VPWS instance descriptive information, RT value, CE ID, maximum planned site number, ID deviation, and interface information from the network administrator.

Configuration Steps

- Configuring CE1

See "Configuring CE1" in basic configuration examples.

- Configuring PE1

# Configure the loopback interface address.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 1.1.1.1 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes so that PEs can ping with ASBRs in the same AS.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 1.1.1.1 0.0.0.0 area 0
Ruijie(config-router)# network 10.10.10.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# exit
```

# Configure the public network tunnel between PEs.

```
Ruijie(config)# interface gigabitEthernet 1/2
Ruijie(config-if-Gigabitethernet 1/2)# ip ref
Ruijie(config-if-Gigabitethernet 1/2)# ip address 10.10.10.1 255.255.255.0
Ruijie(config-if-Gigabitethernet 1/2)# mpls ip
Ruijie(config-if-Gigabitethernet 1/2)# label-switching
Ruijie(config-if-Gigabitethernet 1/2)# exit
```

# Configure the L2VPN address family.

```
Ruijie(config)# router bgp 100
Ruijie(config-router)# neighbor 2.2.2.2 remote-as 100
Ruijie(config-router)# neighbor 2.2.2.2 update-source loopback 0
Ruiije(config-router)# address-family l2vpn vpws
Ruijie(config-router-af)# neighbor 2.2.2.2 activate
Ruijie(config-router-af)# neighbor 2.2.2.2 send-community extended
Ruijie(config-router-af)# exit
```

# Configure a VFI instance.

```
Ruijie# configure terminal
Ruijie(config)# l2 vfi vpws-1 vpnid 1 point-to-point
Ruijie(config-vfi)# rd 100:1
Ruijie(config-vfi)# encapsulation mpls ethernet
Ruijie(config-vfi)# route-target both 10000:1
Ruijie(config-vfi)# site-id 1
Ruijie(config-vfi-site)# xconnect interface gigabitEthernet 1/1 remote-ce-id 2
Ruijie(config-vfi-site)#exit-site-mode
Ruijie(config-vfi)#exit
```

# Configure the interface between CEs and PEs and specify the remote CE ID.

```
Ruijie(config)# interface gigabitEthernet 1/1
Ruijie(config-if-GigabitEthernet 1/1)# ip ref
Ruijie(config-if-GigabitEthernet 1/1)# exit
```

- Configuring ASBR1:

# Configure the loopback interface address.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 2.2.2.2 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF protocol, establish public network routes so that PEs can ping with ASBRs in the same domain.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 2.2.2.2 0.0.0.0 area 0
Ruijie(config-router)# network 10.10.10.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# exit
```

# Configure the public network tunnel between PEs.

```
Ruijie(config)# interface gigabitEthernet 1/1
Ruijie(config-if-Gigabitethernet 1/1)# ip ref
Ruijie(config-if-Gigabitethernet 1/1)# ip address 10.10.10.2 255.255.255.0
Ruijie(config-if-Gigabitethernet 1/1)# mpls ip
Ruijie(config-if-Gigabitethernet 1/1)# label-switching
Ruijie(config-if-Gigabitethernet 1/1)# exit
```

# Configure the L2VPN address family.

```
Ruijie(config)# router bgp 100
Ruijie(config-router)# neighbor 1.1.1.1 remote-as 100
Ruijie(config-router)# neighbor 1.1.1.1 update-source loopback 0
Ruiije(config-router)# address-family l2vpn vpws
Ruijie(config-router-af)# neighbor 1.1.1.1 activate
Ruijie(config-router-af)# exit
```

# Configure a VPWS instance.

```
Ruijie# configure terminal
Ruijie(config)# l2 vfi vpws-1 vpnid 2 point-to-point
Ruijie(config-vfi)# rd 100:1
Ruijie(config-vfi)# encapsulation mpls ethernet
Ruijie(config-vfi)# route-target both 10000:1
Ruijie(config-vfi)# site-id 2
Ruijie(config-vfi-site)# xconnect interface gigabitEthernet 1/2 remote-ce-id 1
Ruijie(config-vfi-site)#exit-site-mode
Ruijie(config-vfi)#exit
```

# Configure the interface between ASBR1 and ASBR2.

```
Ruijie(config)# interface gigabitEthernet 1/2
Ruijie(config-if-GigabitEthernet 1/2)# ip ref
Ruijie(config-if-GigabitEthernet 1/2)# exit
```

- Configure ASBR2:

# Configure the loopback interface address.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 3.3.3.3 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF protocol, establish public network routes so that PEs can ping with ASBRs in the same domain.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 3.3.3.3 0.0.0.0 area 0
Ruijie(config-router)# network 20.20.20.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# exit
```

#Configure the public network tunnel between PEs.

```
Ruijie(config)# interface gigabitEthernet 1/2
Ruijie(config-if-Gigabitethernet 1/2)# ip ref
Ruijie(config-if-Gigabitethernet 1/2)# ip address 20.20.20.2 255.255.255.0
Ruijie(config-if-Gigabitethernet 1/2)# mpls ip
Ruijie(config-if-Gigabitethernet 1/2)# label-switching
Ruijie(config-if-Gigabitethernet 1/2)# exit
```

# Configure the L2VPN address family.

```
Ruijie(config)# router bgp 200
Ruijie(config-router)# neighbor 4.4.4.4 remote-as 200
Ruijie(config-router)# neighbor 4.4.4.4 update-source loopback 0
Ruiije(config-router)# address-family l2vpn vpws
Ruijie(config-router-af)# neighbor 4.4.4.4 activate
Ruijie(config-router-af)# neighbor 4.4.4.4 send-community extended
Ruijie(config-router-af)# exit
```

# Configure a VPWS instance.

```
Ruijie# configure terminal
```

```
Ruijie(config)# l2 vfi vpws-1 vpnid 1 point-to-point
Ruijie(config-vfi)# rd 200:1
Ruijie(config-vfi)# encapsulation mpls ethernet
Ruijie(config-vfi)# route-target both 10000:1
Ruijie(config-vfi)# site-id 3
Ruijie(config-vfi-site)# xconnect interface gigabitEthernet 1/1 remote-ce-id 4
Ruijie(config-vfi-site)#exit-site-mode
Ruijie Networks(config-vfi)#exit
```

# Configure the interface between ASBR1 and ASBR2.

```
Ruijie(config)# interface gigabitEthernet 1/1
Ruijie(config-if-GigabitEthernet 1/1)# ip ref
Ruijie(config-if-GigabitEthernet 1/1)# exit
```

- Configuring PE2:

# Configure the loopback interface address.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 4.4.4.4 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes so that PEs can ping with ASBRs in the same AS.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 4.4.4.4 0.0.0.0 area 0
Ruijie(config-router)# network 20.20.20.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# exit
```

# Configure the public network tunnel between PEs.

```
Ruijie(config)# interface gigabitEthernet 1/1
Ruijie(config-if-Gigabitethernet 1/1)# ip ref
Ruijie(config-if-Gigabitethernet 1/1)# ip address 20.20.20.1 255.255.255.0
Ruijie(config-if-Gigabitethernet 1/1)# mpls ip
Ruijie(config-if-Gigabitethernet 1/1)# label-switching
Ruijie(config-if-Gigabitethernet 1/1)# exit
```

# Configure the L2VPN address family.

```
Ruijie(config)# router bgp 200
Ruijie(config-router)# neighbor 3.3.3.3 remote-as 200
Ruijie(config-router)# neighbor 3.3.3.3 update-source loopback 0
Ruiije(config-router)# address-family l2vpn vpws
Ruijie(config-router-af)# neighbor 3.3.3.3 activate
Ruijie(config-router-af)# neighbor 3.3.3.3 send-community extended
Ruijie(config-router-af)# exit
```

# Configure a VPWS instance.

```
Ruijie# configure terminal
Ruijie(config)# l2 vfi vpws-1 vpnid 1 autodiscovery
Ruijie(config-vfi)# rd 200:1
Ruijie(config-vfi)# signal bgp
Ruijie(config-vfi)# encapsulation mpls ethernet
Ruijie(config-vfi)# route-target both 10000:1
Ruijie(config-vfi)# site-id 4
Ruijie(config-vfi-site)# xconnect interface gigabitEthernet 1/2 remote-ce-id 3
Ruijie(config-vfi-site)#exit-site-mode
Ruijie(config-vfi)#exit
```

# Configure the interface between PE2 and a CE.

```
Ruijie(config)# interface gigabitEthernet 1/2
Ruijie(config-if-GigabitEthernet 1/2)# ip ref
Ruijie(config-if-GigabitEthernet 1/2)# exit
```

- Configuring CE2:

See "Configuring CE2" in basic configuration examples.

Verification

After the configuration, CE1 can ping with CE2.

After completing the configuration of Kompella VPWS, use the following commands to check the operation of VPWS.

Command	Function
Ruijie# **show bgp l2vpn vpws all**	Displays all the VPWS information.
Ruijie# **show mpls l2transport vc** [*vc_id* [*ip-address*]] | [**interface** *interface_name*] [**detail**]	Displays information about the PW (including VPWS PW and VPLS PW).
Ruijie# **show bgp l2vpn** { **vpls | vpws** } **all connections** [**neighbor** *address*] | [**interface** *interface_name*] | [**site-i**d *id*] [**detail**]	Displays PW information.
Ruijie# **show mpls vfi** [*name*]	Displays all the configured or specified VFI information.

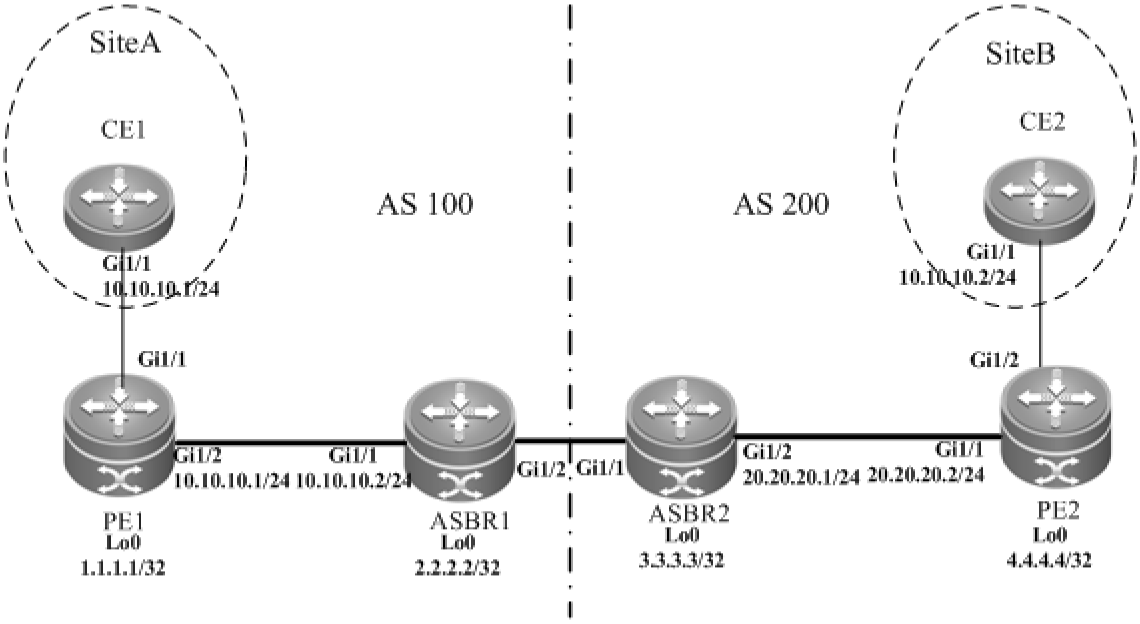
Option C Inter-AS VPWS

Networking Requirements

- CEs of customer S in site A and site B are connected with each other through the carrier's PE1 in AS 100 and PE2 in AS 200, realizing layer-2 point-to-point connection.
- PE1 and PE2 are in different ASs and can automatically detect PE devices involved in the VFI instance.
- ASBR is not responsible for maintaining VPWS label block messages.
- VPWS label block messages are directly switched between PEs.

Networking Topology

Figure 68 Kompella VPWS Option C inter-AS networking topology



The above figure shows the structure of Kompella VPWS Option C Inter-AS networking topology. Customer S's CE devices in Site A and Site B are connected to each other through PE1 in AS100 and PE2 in AS200, thus realizing layer-2 point-to-point connection.

Configuration Tips

Before configuring Kompella VPWS, complete the following tasks:

- Run IGP in the carrier's network to realize connection between PE and ASBR devices in the same AS.
- Establish a public network tunnel between PE and ASBR devices in the same AS and enable MPLS on the ASBR interface.
- Establish IBGP between PE and ASBR in the same AS.
- Establish EBGP between ASBR devices and enable send-label.
- Obtain Kompella VPWS configuration information including VPWS instance descriptive information, RT value, CE, planned site number, ID deviation, and interface information from the network administrator.

Configuration Steps

- Configuring CE1

See "Configuring CE1" in basic configuration examples.

- Configuring PE1

# Configure the loopback interface address.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 1.1.1.1 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 1.1.1.1 0.0.0.0 area 0
Ruijie(config-router)# network 10.10.10.0 0.0.0.255 area 0
Ruijie(config-router)# exit
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# exit
```

# Configure the public network tunnel between PEs.

```
Ruijie(config)# interface gigabitEthernet 1/2
Ruijie(config-if-Gigabitethernet 1/2)# ip ref
Ruijie(config-if-Gigabitethernet 1/2)# ip address 10.10.10.1 255.255.255.0
Ruijie(config-if-Gigabitethernet 1/2)# mpls ip
Ruijie(config-if-Gigabitethernet 1/2)# label-switching
Ruijie(config-if-Gigabitethernet 1/2)# exit
```

# Configure the L2VPN address family.

```
Ruijie(config)# router bgp 100
Ruijie(config-router)# neighbor 4.4.4.4 remote-as 200
Ruijie(config-router)# neighbor 4.4.4.4 update-source loopback 0
Ruijie(config-router)# neighbor 4.4.4.4  ebgp-multihop
Ruijie(config-router)# address-family ipv4
Ruijie(config-router-af)# no neighbor 4.4.4.4  activate
Ruijie(config-router-af)# exit
Ruiije(config-router)# address-family l2vpn vpws
Ruijie(config-router-af)# neighbor 4.4.4.4 activate
Ruijie(config-router-af)# neighbor 4.4.4.4 send-community extended
Ruijie(config-router-af)# exit
```

# Configure a VPWS instance.

```
Ruijie# configure terminal
Ruijie(config)# l2 vfi vpws-1 vpnid 1 point-to-point
Ruijie(config-vfi)# rd 100:1
Ruijie(config-vfi)# encapsulation mpls ethernet
Ruijie(config-vfi)# route-target both 10000:1
Ruijie(config-vfi)# site-id 1
Ruijie(config-vfi-site)# xconnect interface gigabitEthernet 1/1 remote-ce-id 2
Ruijie(config-vfi-site)#exit-site-mode
Ruijie(config-vfi)#exit
```

# Configure the interface that connects CEs.

```
Ruijie(config)# interface gigabitEthernet 1/1
Ruijie(config-if-GigabitEthernet 1/1)# ip ref
Ruijie(config-if-GigabitEthernet 1/1)# exit
```

- Configuring ASBR1

# Configure the loopback interface address.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 2.2.2.2 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# redistribute bgp subnets
Ruijie(config-router)# network 2.2.2.2 0.0.0.0 area 0
Ruijie(config-router)# network 10.10.10.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# advertise-labels for bgp-routes
Ruijie(config-mpls-router)# exit
```

# Configure the public network tunnel between PEs.

```
Ruijie(config)# interface gigabitEthernet 1/1
Ruijie(config-if-Gigabitethernet 1/1)# ip ref
Ruijie(config-if-Gigabitethernet 1/1)# ip address 10.10.10.2 255.255.255.0
Ruijie(config-if-Gigabitethernet 1/1)# mpls ip
Ruijie(config-if-Gigabitethernet 1/1)# label-switching
Ruijie(config-if-Gigabitethernet 1/1)# exit
```

# Configure the interface that connects ASBR.

```
Ruijie(config)# interface gigabitEthernet 1/2
Ruijie(config-if-Gigabitethernet 1/2)# ip ref
Ruijie(config-if-Gigabitethernet 1/2)# ip address 192.168.1.1 255.255.255.252
Ruijie(config-if-Gigabitethernet 1/2)# label-switching
Ruijie(config-if-Gigabitethernet 1/2)# exit
```

# Configure ASBR to allocate labels for PEs' routes.

```
Ruijie(config)# router bgp 100
Ruijie(config-router)# neighbor 192.168.1.2 remote-as 200
Ruijie(config-router)# address-family ipv4
Ruijie(config-router-af)# neighbor 192.168.1.2 send-label
Ruijie(config-router-af)# network 1.1.1.1 mask 255.255.255.255
Ruijie(config-router-af)# end
```

- Configuring ASBR2

# Configure the loopback interface address.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 3.3.3.3 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes.

```
Ruijie(config)# router ospf 20
Ruijie(config-router)# redistribute bgp subnets
Ruijie(config-router)# network 3.3.3.3 0.0.0.0 area 0
Ruijie(config-router)# network 20.20.20.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# advertise-labels for bgp-routes
Ruijie(config-mpls-router)# exit
```

# Configure the public network tunnel between PEs.

```
Ruijie(config)# interface gigabitEthernet 1/2
Ruijie(config-if-Gigabitethernet 1/2)# ip ref
Ruijie(config-if-Gigabitethernet 1/2)# ip address 20.20.20.2 255.255.255.0
Ruijie(config-if-Gigabitethernet 1/2)# mpls ip
Ruijie(config-if-Gigabitethernet 1/2)# label-switching
Ruijie(config-if-Gigabitethernet 1/2)# exit
```

# Configure the interface that connects ASBR.

```
Ruijie(config)# interface gigabitEthernet 1/1
Ruijie(config-if-Gigabitethernet 1/1)# ip ref
Ruijie(config-if-Gigabitethernet 1/1)# ip address 192.168.1.1 255.255.255.252
Ruijie(config-if-Gigabitethernet 1/1)# label-switching
Ruijie(config-if-Gigabitethernet 1/1)# exit
```

# Configure ASBR to allocate labels for PEs' routes.

```
Ruijie(config)# router bgp 200
Ruijie(config-router)# neighbor 192.168.1.1 remote-as 100
Ruijie(config-router)# address-family ipv4
Ruijie(config-router-af)# neighbor 192.168.1.1  send-label
Ruijie(config-router-af)# network 4.4.4.4 mask 255.255.255.255
Ruijie(config-router-af)# end
```

- Configuring PE2

# Configure the loopback interface address.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 4.4.4.4 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes.

```
Ruijie(config)# router ospf 20
Ruijie(config-router)# network 4.4.4.4 0.0.0.0 area 0
Ruijie(config-router)# network 20.20.20.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# exit
```

#Configure the public network tunnel between PEs.

```
Ruijie(config)# interface gigabitEthernet 1/1
Ruijie(config-if-Gigabitethernet 1/1)# ip ref
Ruijie(config-if-Gigabitethernet 1/1)# ip address 20.20.20.1 255.255.255.0
Ruijie(config-if-Gigabitethernet 1/1)# mpls ip
Ruijie(config-if-Gigabitethernet 1/1)# label-switching
Ruijie(config-if-Gigabitethernet 1/1)# exit
```

# Configure the L2VPN address family.

```
Ruijie(config)# router bgp 100
Ruijie(config-router)# neighbor 1.1.1.1 remote-as 100
Ruijie(config-router)# neighbor 1.1.1.1 update-source loopback 0
Ruijie(config-router)# neighbor  1.1.1.1  ebgp-multihop
Ruijie(config-router)# address-family ipv4
Ruijie(config-router-af)# no neighbor  1.1.1.1 activate
Ruijie(config-router-af)# exit
Ruiije(config-router)# address-family l2vpn vpws
Ruijie(config-router-af)# neighbor 1.1.1.1 activate
Ruijie(config-router-af)# neighbor 1.1.1.1 send-community extended
Ruijie(config-router-af)# exit
```

# Configure a Kompella VPWS instance.

```
Ruijie# configure terminal
Ruijie(config)# l2 vfi vpws-1 vpnid 1 point-to-point
Ruijie(config-vfi)# rd 200:1
Ruijie(config-vfi)# encapsulation mpls ethernet
Ruijie(config-vfi)# route-target both 10000:1
Ruijie(config-vfi)# mtu 1500
Ruijie(config-vfi)# site-id 2
Ruijie(config-vfi-site)# xconnect interface gigabitEthernet 1/2 remote-ce-id 1
Ruijie(config-vfi-site)#exit-site-mode
Ruijie(config-vfi)#exit
```

# Configure the interface between PEs and CEs.

```
Ruijie(config)# interface gigabitEthernet 1/2
Ruijie(config-if-GigabitEthernet 1/2)# ip ref
Ruijie(config-if-GigabitEthernet 1/2)# exit
```

- Configuring CE2

See "Configuring CE2" in basic configuration examples.

Verification

After the configuration, CE1 can ping with CE2.

After completing the configuration of Kompella VPLS, use the following commands to check the operation of VPWS:

Command	**Function**
Ruijie# **show bgp l2vpn vpws all**	Displays all the VPWS information.
Ruijie# **show mpls l2transport vc** [*vc_id* [*ip-address*]] | [**interface** *interface_name*] [**detail**]	Displays information about the PW (including VPWS PW and VPLS PW).
Ruijie# **show bgp l2vpn** { **vpls | vpws** } **all connections** [**neighbor** *address*] | [**interface** *interface_name*] | [**site-id** *id*] [**detail**]	Displays PW information.
Ruijie# **show mpls vfi** [*name*]	Displays all the configured or specified VFI information.

Examples of Maintaining PW Connectivity of VPWS

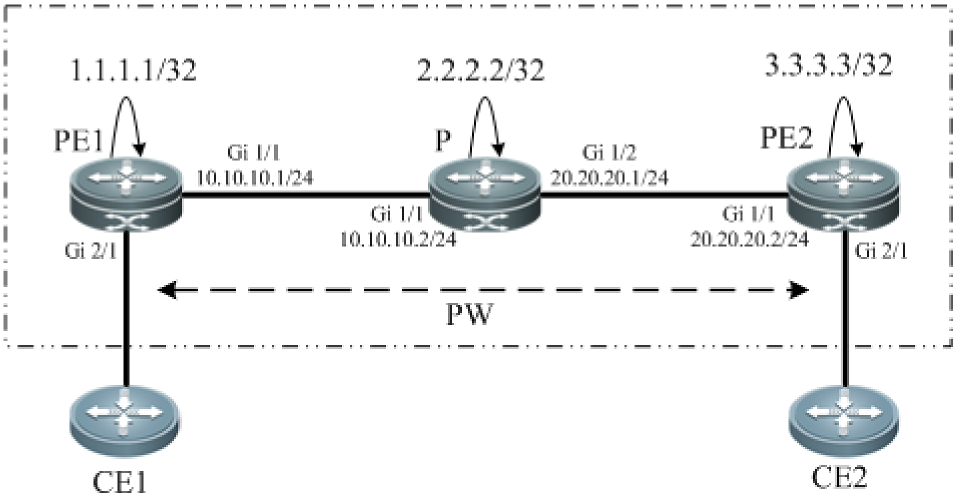
Verifying PW Connectivity and Fault of Martini VPWS

Networking Requirements

- The CE devices play the role of the customer network to connect to the PE devices via the AC.
- PE1 and PE2 establish a PW between them via LDP.
- PE1, P, and PE2 set up the public tunnel and LSP path via LDP.

Network Topology

Figure 33



Configuration Key Points

The configuration key points for PE1, P, and PE2 are as follows:

- Configure the IP address of the interface of each device and configure the Open Shortest Path First (OSPF) protocol.
- Configure the global MPLS capability and the interface MPLS capability of each device.
- Configure LDP to enable the network to forward MPLS traffic.
- Configure PWE3.
- Check the PW connectivity.
- Locate the PW fault.

Configuration Steps

(1) Configure the IP address of the interface of each node and configure the OSPF protocol.

#Configure PE1.

```
Ruijie(config)#interface loopback 0
Ruijie(config-if-Loopback 0)#ip address 1.1.1.1 255.255.255.255
Ruijie(config-if-Loopback 0)#exit
PE1 config-if-GigabitEthernet 1/1)#
Ruijie(config-if-GigabitEthernet 1/1)#ip address 10.10.10.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/1)#exit
Ruijie(config)#router ospf 10
Ruijie(config-router)#network 1.1.1.1 0.0.0.0 area 0
Ruijie(config-router)#network 10.10.10.0 0.0.0.255 area 0
```

Ruijie(config-router)#exit#Configure P.

```
Ruijie(config)#interface loopback 0
Ruijie(config-if-Loopback 0)#ip address 2.2.2.2 255.255.255.255
Ruijie(config-if-Loopback 0)#exit
Ruijie(config)#interface gigabitEthernet 1/1
Ruijie(config-if-GigabitEthernet 1/1)#ip address 10.10.10.2 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/1)#exit
Ruijie(config)#interface gigabitEthernet 1/2
Ruijie(config-if-GigabitEthernet 1/2)#ip address 20.20.20.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/2)#exit
Ruijie(config)#router ospf 10
Ruijie(config-router)#network 2.2.2.2 0.0.0.0 area 0
Ruijie(config-router)#network 10.10.10.0 0.0.0.255 area 0
Ruijie(config-router)#network 20.20.20.0 0.0.0.255 area 0
Ruijie(config-router)#exit
```

#Configure PE2.

```
Ruijie(config)#interface loopback 0
Ruijie(config-if-Loopback 0)#ip address 3.3.3.3 255.255.255.255
Ruijie(config-if-Loopback 0)#exit
Ruijie(config)#interface gigabitEthernet 1/1
Ruijie(config-if-GigabitEthernet 1/1)#ip address 20.20.20.2 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/1)#exit
Ruijie(config)#router ospf 10
Ruijie(config-router)#network 3.3.3.3 0.0.0.0 area 0
Ruijie(config-router)#network 20.20.20.0 0.0.0.255 area 0
Ruijie(config-router)#exit
```

(2) Configure the global MPLS capability and the interface MPLS capability of each node and enable the LDP function on the interfaces.

#Configure PE1.

```
Ruijie(config)#mpls ip
Ruijie(config)#mpls router ldp
Ruijie(config-mpls-router)#ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)#neighbor 3.3.3.3.
Ruijie(config-mpls-router)#exit
Ruijie(config)#interface gigabitEthernet 1/1
Ruijie(config-if-GigabitEthernet 1/1)#mpls ip
Ruijie(config-if-GigabitEthernet 1/1)#label-switching
Ruijie(config-if-GigabitEthernet 1/1)#exit
```

#Configure P.

```
Ruijie(config)#mpls ip
Ruijie(config)#mpls router ldp
Ruijie(config-mpls-router)#ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)#exit
Ruijie(config)#interface gigabitEthernet 1/1
Ruijie(config-if-GigabitEthernet 1/1)#mpls ip
Ruijie(config-if-GigabitEthernet 1/1)#label-switching
Ruijie(config-if-GigabitEthernet 1/1)#exit
Ruijie(config)#interface gigabitethernet 1/2
Ruijie(config-if-GigabitEthernet 1/2)#mpls ip
Ruijie(config-if-GigabitEthernet 1/2)#label-switching
Ruijie(config-if-GigabitEthernet 1/2)#exit
```

#Configure PE2.

```
Ruijie(config)#mpls ip
Ruijie(config)#mpls router ldp
Ruijie(config-mpls-router)#ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)#neighbor 1.1.1.1
Ruijie(config-mpls-router)#exit
Ruijie(config)#interface gigabitEthernet 1/1
Ruijie(config-if-GigabitEthernet 1/1)#mpls ip
Ruijie(config-if-GigabitEthernet 1/1)#label-switching
Ruijie(config-if-GigabitEthernet 1/1)#exit
```

(3) Configure VPWS instances between PE1 and PE2 and enable CE1 and CE2 to access the VPWS.

#Configure PE1.

```
Ruijie(config)#interface gigabitethernet 2/1
Ruijie(config-if-Gigabitethernet 2/1)#xconnect 3.3.3.3 100  encapsulation mpls ethernet
Ruijie(config-if-Gigabitethernet 2/1)#exit
```

#Configure PE2.

```
Ruijie(config)#interface gigabitethernet 2/1
Ruijie(config-if-Gigabitethernet 2/1)#xconnect 1.1.1.1 100  encapsulation mpls ethernet
Ruijie(config-if-Gigabitethernet 2/1)#exit
```

(4) Verify the PW connectivity.

```
Ruijie#ping mpls pseudowire 3.3.3.3 100 label-alert
Sending 5, 112-byte MPLS Echoes on Pseudowire, peer address 3.3.3.3, VC ID 100,
    timeout is 2 seconds, send interval is 0 msec:

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
 'L'-labeled output interface,'B'-unlabeled output interface,
 'D'-DS Map mismatch,'F'-no FEC mapping,'f'-FEC mismatch,
 'M'-malformed request,'m'-unsupported tlvs,'N'-no label entry,
 'P'-no rx intf label prot,'p'-premature termination of LSP,
 'R'-transit router,'I'-unknown upstream index,
 'X'-unknown return code,'x'-return code 0
Press Ctrl+C to break.
!!!!!
Success rate is 100 percent(5/5),round-trip min/avg/max=20/36/60 ms
```

(5) Locate the PW fault.

```
Ruijie#traceroute mpls pseudowire 3.3.3.3 100 segment 1 ttl-expiry
Traceing Pseudowire Label Switched Path, peer address 3.3.3.3, VC ID 100, timeout is 2 seconds

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Press Ctrl+C to break.
  0 10.10.10.2     MRU 1500 [Labels: 1025 Exp: 0]
! 1 20.20.20.2     60 ms [Labels: 1025 Exp: 0]
local 1.1.1.1 remote 3.3.3.3 vc id 100
```

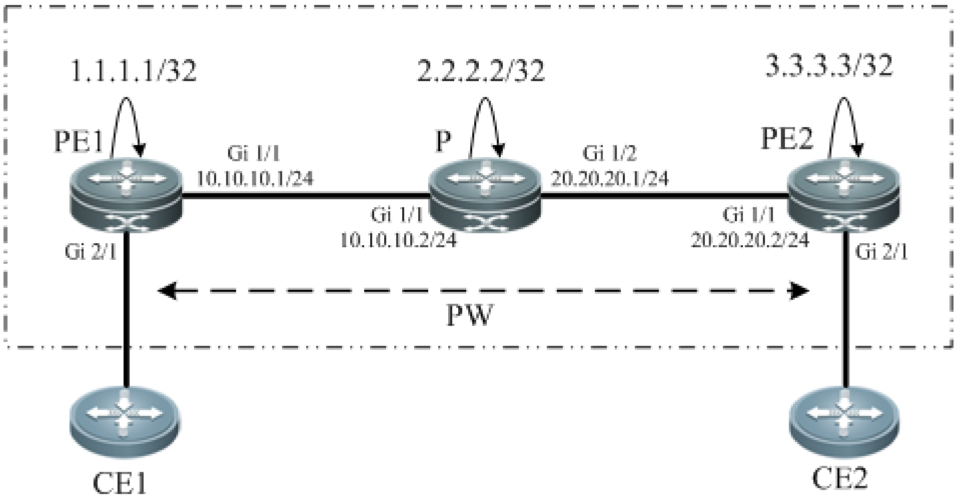
Verifying PW Connectivity of Kompella VPWS

Networking Requirements

As shown in Figure 3-25, PE1, P, and PE2 belong to the same MPLS domain, and CE1 and CE2 belong to the same VPN instance. PE1 and PE2 establish a PW between them and use the ping or traceroute command to check the PW connectivity to ensure services on the links.

Network Topology

Figure 34



Configuration Key Points

The configuration key points for each node are as follows:

(1). Configure the IP address of the interface of each node and configure the OSPF protocol.

(2). Configure the global MPLS capability and the interface MPLS capability of each node.

(3). Enable the L2VPN on PE1 and PE2 and create a VC connection.

(4). Check the PW connectivity.

(5). Locate the PW fault.

Configuration Steps

(1) Configure the IP address of the interface of each node and configure the OSPF protocol.

#Configure PE1.

```
Ruijie(config)#interface loopback 0
Ruijie(config-if-Loopback 0)#ip address 1.1.1.1 255.255.255.255
Ruijie(config-if-Loopback 0)#exit
Ruijie(config-if-GigabitEthernet 1/1)#ip address 10.10.10.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/1)#exit
Ruijie(config)#router ospf 10
Ruijie(config-router)#network 1.1.1.1 0.0.0.0 area 0
Ruijie(config-router)#network 10.10.10.0 0.0.0.255 area 0
Ruijie(config-router)#exit
```

#Configure P.

```
Ruijie(config)#interface loopback 0
Ruijie(config-if-Loopback 0)#ip address 2.2.2.2 255.255.255.255
Ruijie(config-if-Loopback 0)#exit
Ruijie(config)#interface gigabitEthernet 1/1
Ruijie(config-if-GigabitEthernet 1/1)#ip address 10.10.10.2 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/1)#exit
Ruijie(config)#interface gigabitEthernet 1/2
Ruijie(config-if-GigabitEthernet 1/2)#ip address 20.20.20.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/2)#exit
Ruijie(config)#router ospf 10
Ruijie(config-router)#network 2.2.2.2 0.0.0.0 area 0
Ruijie(config-router)#network 10.10.10.0 0.0.0.255 area 0
Ruijie(config-router)#network 20.20.20.0 0.0.0.255 area 0
Ruijie(config-router)#exit
```

#Configure PE2.

```
Ruijie(config)#interface loopback 0
Ruijie(config-if-Loopback 0)#ip address 3.3.3.3 255.255.255.255
Ruijie(config-if-Loopback 0)#exit
Ruijie(config)#interface gigabitEthernet 1/1
Ruijie(config-if-GigabitEthernet 1/1)#ip address 20.20.20.2 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/1)#exit
Ruijie(config)#router ospf 10
Ruijie(config-router)#network 3.3.3.3 0.0.0.0 area 0
Ruijie(config-router)#network 20.20.20.0 0.0.0.255 area 0
Ruijie(config-router)#exit
```

(2) Configure the global MPLS capability and the interface MPLS capability of each node and enable the LDP function on the interfaces.

#Configure PE1.

```
Ruijie(config)#mpls ip
Ruijie(config)#mpls router ldp
Ruijie(config-mpls-router)#ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)#exit
Ruijie(config)#interface gigabitEthernet 1/1
Ruijie(config-if-GigabitEthernet 1/1)#mpls ip
Ruijie(config-if-GigabitEthernet 1/1)#label-switching
Ruijie(config-if-GigabitEthernet 1/1)#exit
```

#Configure P.

```
Ruijie(config)#mpls ip
Ruijie(config)#mpls router ldp
Ruijie(config-mpls-router)#ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)#exit
Ruijie(config)#interface gigabitEthernet 1/1
Ruijie(config-if-GigabitEthernet 1/1)#mpls ip
Ruijie(config-if-GigabitEthernet 1/1)#label-switching
Ruijie(config-if-GigabitEthernet 1/1)#exit
Ruijie(config)#interface gigabitEthernet 1/2
Ruijie(config-if-GigabitEthernet 1/2)#mpls ip
Ruijie(config-if-GigabitEthernet 1/2)#label-switching
Ruijie(config-if-GigabitEthernet 1/2)#exit
```

#Configure PE2.

```
Ruijie(config)#mpls ip
Ruijie(config)#mpls router ldp
Ruijie(config-mpls-router)#ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)#exit
Ruijie(config)#interface gigabitEthernet 1/1
Ruijie(config-if-GigabitEthernet 1/1)#mpls ip
Ruijie(config-if-GigabitEthernet 1/1)#label-switching
Ruijie(config-if-GigabitEthernet 1/1)#exit
```

(3) Configure a L2VPN address family.

#Configure PE1.

```
Ruijie(config)#router bgp 100
Ruijie(config-router)#neighbor 3.3.3.3 remote-as 100
Ruijie(config-router)#neighbor 3.3.3.3 update-source loopback 0
Ruiije(config-router)#address-family l2vpn vpws
Ruijie(config-router-af)#neighbor 3.3.3.3 activate
Ruijie(config-router-af)#neighbor 3.3.3.3 send-community extended
Ruijie(config-router-af)#exit
```

#Configure PE2.

```
Ruijie(config)#router bgp 100
Ruijie(config-router)#neighbor 1.1.1.1 remote-as 100
Ruijie(config-router)#neighbor 1.1.1.1 update-source loopback 0
Ruiije(config-router)#address-family l2vpn vpws
Ruijie(config-router-af)#neighbor 1.1.1.1 activate
Ruijie(config-router-af)#neighbor 1.1.1.1 send-community extended
Ruijie(config-router-af)#exit
```

(4) Configure a VFI instance.

#Configure PE1.

```
Ruijie(config)#l2 vfi vpws-1 vpnid 1 point-to-point
Ruijie(config-vfi)#rd 1:1
Ruijie(config-vfi)#encapsulation mpls ethernet
Ruijie(config-vfi)#route-target both 10000:1
Ruijie(config-vfi)#site-id 1
Ruijie(config-vfi-site)#xconnect interface gigabitEthernet 2/1 remote-ce-id 2
Ruijie(config-vfi-site)#exit-site-mode
Ruijie(config-vfi)#exit
```

#Configure PE2.

```
Ruijie(config)#l2 vfi vpws-1 vpnid 1 point-to-point
Ruijie(config-vfi)#rd 1:1
Ruijie(config-vfi)#encapsulation mpls ethernet
Ruijie(config-vfi)#route-target both 10000:1
Ruijie(config-vfi)#site-id 2
Ruijie(config-vfi-site)#xconnect interface gigabitEthernet 2/1 remote-ce-id 1
Ruijie(config-vfi-site)#exit-site-mode
Ruijie(config-vfi)#exit
```

(5) Verify the PW connectivity.

```
Ruijie#ping mpls pseudowire kompella vpws-1 1 2 rd 1:1 label-alert
Sending 5, 112-byte MPLS Echoes on Pseudowire, VFI name vpws-1, local site id, 1, remote site
id 2,
    timeout is 2 seconds, send interval is 0 msec:

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
 'L'-labeled output interface,'B'-unlabeled output interface,
 'D'-DS Map mismatch,'F'-no FEC mapping,'f'-FEC mismatch,
 'M'-malformed request,'m'-unsupported tlvs,'N'-no label entry,
 'P'-no rx intf label prot,'p'-premature termination of LSP,
 'R'-transit router,'I'-unknown upstream index,
 'X'-unknown return code,'x'-return code 0
Press Ctrl+C to break.
!!!!!
Success rate is 100 percent(5/5),round-trip min/avg/max=20/36/60 ms
```

(6) Locate the PW fault.

```
Ruijie#traceroute mpls pseudowire kompella vpws-1 1 2 rd 1:1 label-alert
Traceing Pseudowire Label Switched Path, VFI name vpws-1, local site id 1, remote site id 2,
timeout is 2 seconds.
Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Press Ctrl+C to break.
  0 10.10.10.2    MRU 1500 [Labels: router-alert/1538 Exp: 0/0]
! 1 20.20.20.2    20 ms
```

VPLS

Introduction to VPLS

The Virtual Private LAN Service (VPLS) is a technology that provides virtual and dedicated Ethernet services on an IP/MPLS network. By using VPLS, you can set up PWs in full mesh mode between PEs to forward encapsulated Layer 2 Ethernet frames between the PEs on the MPLS network. In this manner, you can create a P2MP Ethernet VPN. With a VPLS VPN, the user Layer 2 devices are connected to each other across the IP/MPLS core network and the core network is like a virtual switch for the user.

Compared with VPWS, VPLS can provide P2MP solutions. Compared with L3VPN, VPLS has the advantages of L2VPN to provide better network scalability and maintenance.

The terms used in this document are as follows:

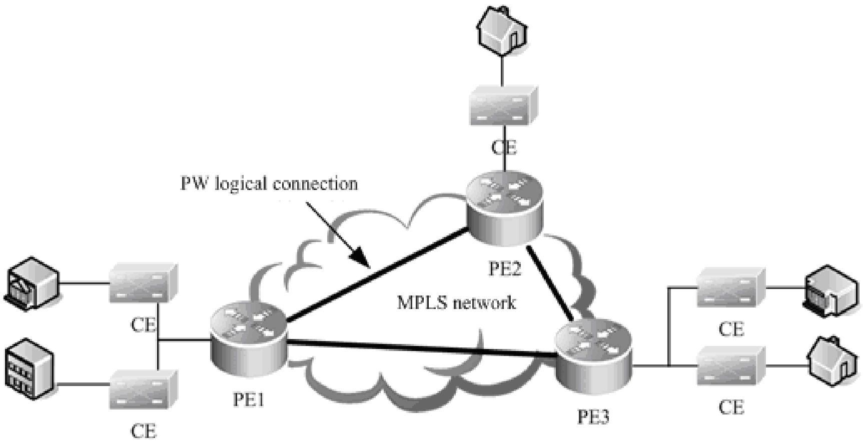
- PW: Pseudo Wire
- VPLS: Virtual Private LAN Service
- H-VPLS: Hierarchical VPLS
- MTU: Multi-Tenant Unit indicates corporate users at office buildings and business districts.
- As small edge devices at gathering places, the MTU is responsible for accessing corporate customers and aggregating users who require VPLS services to a PE through VCs.
- Spoke Connection: indicates the connection of a U-PE to an N-PE in H-VPLS. You can use a PW or QinQ.
- Spoke PW: indicates the Spoke connection between an N-PE and a U-PE when a PW is adopted, or the PW between a CE and PE when a user accesses the CE through PWs.

VPLS Network Structure

Basic VPLS

The following figure shows a typical VPLS model.

Figure 69



As shown in the VPLS networking in the preceding figure, PW logical connections are set up between PEs and the CEs are connected to the PEs enabled with VPLS. These VPLS PEs form P2MP services for the CEs. In this manner, the PEs are like a Layer 2 switch for the CEs to connect to.

The VPLS model in the dual-IGP instance has the following defects:

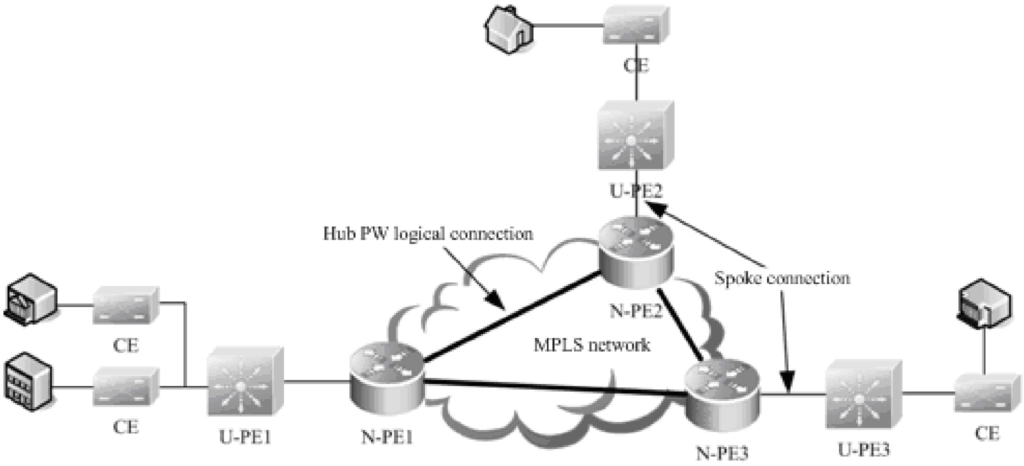
- Packets forwarded on the PWs between PEs are horizontally partitioned to avoid loops. That is, the packets received from a Hub PW are not forwarded to the Hub PW. Therefore, PWs must be set up between every two PEs. This incurs large system overheads and is not applicable to networks of large scales.
- The broadcast and multicast packets must be replicated at every PW, leading to low efficiency.

To address these problems, the hierarchical VPLS is introduced.

Hierarchical VPLS

The following figure shows a typical hierarchical VPLS (H-VPLS) model.

Figure 70



H-VPLS classifies the VPLS network into layers and sets up Hub PWs only between N-PEs on the core network. This addresses the problems on a common VPLS network. The Spoke connections between U-PEs and N-PEs are formed through PWs or QinQ tunnels. The U-PE can support VPWS or QinQ and is not required to support VPLS.

Using H-VPLS can reduce the burden on the VPLS core devices, decrease the overheads of the signaling protocol, reduce the number of packets that are replicated, and greatly strengthen the scalability of the VPLS network.

VPLS Signaling Protocol

VPLSs can be divided into Martini and Kompella VPLSs according to different VPLS signaling protocols. These two types of signaling are defined respectively in RFC4762 and RFC4761 of IETF. The Martini VPLS is based on the LDP signaling protocol and Kompella VPLS is based on the BGP signaling protocol. Given the BGP protocol's characteristics, such as the route reflector's characteristics, the Kompella VPLS can reduce the full inter-connection of BGP sessions, therefore facilitating the expansion of capacity. The Martini VPLS applies to small-sized and simple environment deployment.

VPLS Inter-AS

Both Martini and Kompella VPLSs can be inter-AS through Option A solution. In the inter-AS VPLS networking environment, the type of link between ASBRs must be the same as the type of VC. A sub-interface must be prepared for each inter-AS VC on ASBR. This solution can be adopted when the number of inter-AS VCs is small.

Option C is another solution to realize inter-AS VPLS. The SP network device only needs external tunnels on PEs in different ASs. ASBR does not maintain inter-AS VPLS information and does not need to prepare an interface for the inter-AS VPLS. VSI information of the VPLS can only be exchanged between PEs, reducing resource consumption without adding configuration tasks. The solution applies when the number of inter-AS VPLSs is large.

Configuring Martini VPLS

Configuring a Public Tunnel Between PEs

You must set up an LSP on the public network to carry VPLS services. To run MPLS on the backbone network, you must enable LDP on Ps and PEs at the same time to set up a public tunnel. This means that you have to configure LDP on MPLS devices and enable MPLS forwarding on each interface. The configuration procedure is as follows:

Command	**Function**
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **mpls ip**	Enables MPLS globally. ⚠ Caution This command is not applicable to switch chip forwarding.
Ruijie(config)# **mpls router ldp**	Enables LDP and enters LDP configuration mode.
Ruijie(config-mpls-router)# **ldp router-id interface loopback** *id* [**force**]	Configures the LDP router ID. The IP address of the loopback interface is generally used as the router ID.
Ruijie(config-mpls-router)# **exit**	Exits LDP configuration mode.
Ruijie(config)# **interface** *type ID*	Enters public network interface configuration mode.
Ruijie(config-if-*type ID*)# **label-switching**	Enables MPLS on the interface at the public network side.
Ruijie(config-if-*type ID*)# **mpls ip**	Enables LDP and MPLS forwarding for the interface.

Ruijie(config-if-type ID)# **ip ref**	The interface's fast forwarding function must be enabled for routers.
Ruijie(config-if-type ID)# **end**	Returns to privileged mode.
Ruijie# **copy running-config startup-config**	Saves the configuration.

Caution

LDP is a topology-driven protocol. To ensure normal working of the LDP, enable IPv4 routing protocols and ensure their normal operations.

Configuring Remote LDP Peers

A PW is set up and maintained by the extended LDP. If other LSRs exist between two PEs, use the extended LDP discovery mechanism to set up a remote LDP session between the PEs and assign PW labels in the session. The procedures for configuring a remote LDP peer and setting up a remote LDP session are as follows:

Command	Function
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **mpls router ldp**	Enables LDP and enters LDP configuration mode.
Ruijie(config-mpls-router)# **neighbor** *ip_address*	Configures remote LDP peers.
Ruijie(config-mpls-router)# **exit**	Exits LDP configuration mode.
Ruijie(config)# **end**	Returns to privileged mode.
Ruijie# **copy running-config startup-config**	Saves the configuration.

Caution

The PW label messages of the LDP are not affected by the LDP label distribution mode or label retention mode. The LDP is forced to work in DU and liberal label retention mode.

Martini VPLS Configuration Example

The **l2 vfi** command is used to create a VPLS instance and enter VPLS mode. The **no l2 vfi** command is used to delete a VPLS instance. When creating a VPLS instance, you must specify a name that is unique on the local device and specify a unique VPN ID. Each VPLS name corresponds to a VPN ID.

Command	Function
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **l2 vfi** *vpls-name* **vpnid** *vpn-id*	Creates a VPLS instance and enters VPLS configuration mode.
Ruijie(config-vfi)# **neighbor** *ip-address* **encapsulation mpls** [**vc-id** *vc-id*] [**hub-vc** | **spoke-vc**] [**ethernet** | **ethernetvlan**]	Configures a VPLS peer. The default PW type is Ethernet. We recommend using the PW type for access interface access and VLAN tunnel interface access on switches and Ethernet interface access on routers. ☑ We recommend setting the PW type as Ethernet VLAN if switchers use Trunk interface access or routers use sub-interface access VPLS service.

Ruijie(config-vpls)# **end**	Returns to privileged mode.
Ruijie# **copy running-config startup-config**	Saves the configuration.

⚠ Caution

The VPLS peer must be unique in the VPLS instance scope. To facilitate management, you must configure VPLS instances of one VPN with the same ID.

A PW's key is the PW ID and the VPLS peer's LSR ID. It must be unique globally, including VPWS PW.

Each configuration of the same peer by using the **neighbor** command will cover the previous one.

The **mtu** parameter and PW type of interfaces on two ends of a PW must be the same. The default PW type of the VPLS is Ethernet. PW IDs on two ends of PEs(that is, the VC IDs) must be the same.

When using the **neighbor** command to specify the address of the VC's peer neighbor, you must use the peer router ID as the peer address and the 32-bit address of the loopback interface as the peer router ID.

☑ For switches, DHCP packets cannot be transmitted transparently after **ip dhcp snooping** is enabled globally.

Configuring User Access VPLS

☑ For switches, each VPLS can bind only one interface. For routers, multiple interfaces can be bound.

Configuring VPLS access mode for switches

A VPLS instance takes effect only after the user of the VPLS instance accesses the VPLS. At present, there are three modes of user access VPLS services:

☑ For switches, a VPLS can bind only one SVI interface because VPLS is supported only by the SVI interface.

⚠ Caution

When the port protection mode is enabled on the AC-end member port of a L2VPN, the port protection mode does not take effect on the member port if the corresponding member port is not a trunk interface.

- VLAN access interface access

The VLAN access interface is applicable to the transmission of user packets that are not encapsulated through 802.1Q (that is, packets without VLAN tags) on VPLS ACs.

Command	Function
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **interface** *type ID*	Enters interface mode of the specified physical interface.
Ruijie(config-if-*type ID*)# **switchport mode access**	Enables the interface to work in access mode.
Ruijie(config-if-*type ID*)# **switchport access** *vlan-id*	Sets the interface as a VLAN member interface.
Ruijie(config-if-*type ID*)# **exit**	Exits interface configuration mode.
Ruijie(config)# **interface vlan** *vlan-id*	Enters VLAN interface mode.
Ruijie(config-if-*type ID*)# **xconnect vfi** *vpls-name*	Binds the VLAN to a VPLS instance.
Ruijie(config-if-*type ID*)# **end**	Returns to privileged mode.
Ruijie# **copy running-config startup-config**	Saves the configuration.

Caution

For the access interface access mode, we recommend setting the PW type to **ethernet**, and the two ends of PW must be set in the same type.

- VLAN trunk interface access

The VLAN trunk interface access is applicable to the transmission of multiple VPLS services on the same physical AC. Each VLAN corresponds to a VPLS instance. The PE determines the VPLS instance for user packets based on their VLAN tags to provide the multiplexing of access interfaces.

Command	**Function**
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **interface** *type ID*	Enters interface mode of the specified physical interface.
Ruijie(config-if-*type ID*)# **switchport mode trunk**	Enables the interface to work in trunk mode.
Ruijie(config-if-*type ID)#* **switchport trunk allow vlan** *vlan-list*	Enables the trunk link to allow VLAN traffic.
Ruijie(config-if-*type ID*)# **exit**	Exits interface configuration mode.
Ruijie(config)# **interface vlan** *vlan-id*	Enters VLAN interface mode.
Ruijie(config-if-*type ID*)# **xconnect vfi** *vpls-name*	Binds the VLAN to a VPLS instance.
Ruijie(config-if-*type ID*)# **end**	Returns to privileged mode.
Ruijie# **copy running-config startup-config**	Saves the configuration.

Caution

For the Trunk interface access mode, we recommend setting the PW type as **ethernetvlan**, and the two ends of PW must be set in the same type.

The L2 VPN service cannot be bound to the Native VLAN of the Trunk interface.

- VLAN tunnel interface access

The VLAN tunnel interface access is applicable to the transmission of user service packets that carry private VLAN tags on the ACs when a user accesses VPLS services. In this mode, the PE forwards all packets received from the VLAN tunnel interface without any changes. This mode requires the VLAN member interfaces between PEs and CEs to work in tunnel mode.

Command	**Function**
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **interface** *type ID*	Enters interface mode of the specified physical interface.
Ruijie(config-if-*type ID*)# **switchport access** *vlan-id*	Sets the interface as a VLAN member interface.
Ruijie(config-if-*type ID*)# **switchport mode dot1q-tunnel**	Enables the interface to work in tunnel mode.
Ruijie(config-if-*type ID*)# **exit**	Exits interface configuration mode.
Ruijie(config)# **interface vlan** *vlan-id*	Enters VLAN interface mode.
Ruijie(config-if-*type ID*)# **xconnect vfi** *vpls-name*	Binds the VLAN to a VPLS instance.
Ruijie(config-if-*type ID*)# **end**	Returns to privileged mode.
Ruijie# **copy running-config startup-config**	Saves the configuration.

Caution

For the VLAN tunnel interface access mode, we recommend setting the PW type as **Ethernet,** and the two ends of PW must be set in the same type.

For the VLAN tunnel interface access mode, only the basic QinQ is supported.

- PW access

The PW access is applicable to user access networks that are enabled with MPLS. In addition, the user is not directly connected to any VPLS PE link. In the H-VPLS model, the PW access mode can be used to access the N-PE if the U-PE has no bridging capability so that the VPLS access service can be provided on the U-PE for the user.

Command	Function
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **l2 vfi** *name*	Enters VPLS configuration mode.
Ruijie(config-vpls)# **neighbor** *ip-address* **encapsulation mpls** [vc-id *vc-id*] **spoke-vc** [**ethernet** | **ethernetvlan**]	Configures a PW for user access.
Ruijie(config-vpls)# **end**	Returns to privileged mode.
Ruijie# **copy running-config startup-config**	Saves the configuration.

For one VPLS instance, you can configure multiple PWs for user access.

Caution

When accessing VPLS services through PWs, you must specify the **spoke-vc** keyword in the **neighbor** command to configure PWs, enable the **label-switching** command on the Spoke PW access interface, and enable fast forwarding for routers.

Configuring VPLS access mode for routers

A created VPLS instance takes effect only after the user configured with the VPLS instance is connected with the link.

- Ethernet interface access

This mode applies when the user service packets transmitted on ACs carry private VLAN tags or do not carry VLAN tags in the case of VPLS service access. In this mode, all packets received by PEs from the interface are forwarded according to the destination MAC address and the private tags are considered part of the data. In such case, PEs provide port-based VPLS service. The address learning mode of the VPLS instance of the bound port is the free mode. MAC address overlapping in the user VLAN is not supported.

Command	Function
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **interface** *type ID*	Enters the specified physical interface.
Ruijie(config-if-type ID)# **ip ref**	The interface's fast forwarding function must be enabled for routers.
Ruijie(config-if-*type ID*)# **xconnect vfi** *vpls-name*	Binds VLAN to the VPLS instance.
Ruijie(config-if-*type ID*)# **end**	Returns to privileged mode.
Ruijie# **copy running-config startup-config**	Saves configuration.

Caution

For the VPLS service of the Ethernet interface access mode, we recommend setting the PW type to **ethernet**, and both ends of the PW in the same PW type.

The same VPLS instance can be bound to different interfaces to realize local connection.

If a VPLS instance is configured on different PEs with Ethernet interface and sub-interface access modes, we recommend setting all PWs of the VPLS instance to the **ethernetvlan** type.

- Sub-interface access

This mode applies when multiple VPLS services are transmitted on a physical AC. Each sub-interface corresponds to a VPLS instance. PE devices can match packets with VPLS instances according to VLAN tags carried by the user packets to provide the multiplexing of access interfaces.

Command	**Function**
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **interface** *type ID*	Enters the specified physical interface.
Ruijie(config-if-*type ID*)# **encapsulation dot1Q**	Sets the VIP encapsulated by the sub-interface.
Ruijie(config-if-type ID)# **ip ref**	The interface's fast forwarding function must be enabled for routers.
Ruijie(config-if-*type ID*)# **xconnect vfi** *vpls-name*	Binds VLAN to the VPLS instance.
Ruijie(config-if-*type ID*)# **end**	Returns to privileged mode.
Ruijie# **copy running-config startup-config**	Saves configuration.

Caution

If a VPLS instance is used to transmit flows of multiple VLANs of the user, MAC address overlapping between VLANs of the user is not supported.

The same VPLS instance can be bound to different interfaces to realize local connection.

For the VPLS service of the sub-interface access mode, we recommend setting the PW type to **ethernetvlan**, and both ends of the PW in the same PW type.

- PW access

This mode applies when the user's access network is the MPLS network and no direct link is set up between the user and VPLS PEs. In the H-VPLS model, the PW access mode can be used to access the N-PE when the U-PE has no bridging capability so that the VPLS access service can be provided on the U-PE for the user.

Command	**Function**
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **l2 vfi** *name*	Enters VPLS configuration mode.
Ruijie(config-vfi)# **neighbor** *ip-address* **encapsulation mpls** [**vc-id** *vc-id*] **spoke-vc** [**ethernet | ethernetvlan**]	Configures the PW to be accessed by user.
Ruijie(config-vfi)# **end**	Returns to privileged mode.
Ruijie# **copy running-config startup-config**	Saves configuration.

Caution

One VPLS instance can be configured with multiple PWs for user access.

When accessing VPLS services through PWs, you must specify the **spoke-vc** keyword in the **neighbor** command to configure PWs.

☑ For routers, the **label-switching** and **ip ref** commands must be enabled on the interface where Spoke PW is located.

Caution

If the Ethernet access mode is adopted, the PW type must be **ethernet**. If the Ethernet sub-interface access mode is adopted, the PW type must be **ethernetvlan**.

Verifying Martini VPLS Configuration

After completing the configuration of the VPLS, use the following commands to check the operation of VPLS.

Command	Function
Ruijie# **show mpls vfi**	Displays VPLS information.
Ruijie# **show mpls l2transport vc** [[*vc_id* [*ip-address*]] | [**interface** *interface-name*]] [**detail**]	Displays information about the PW (including VPWS PW and VPLS PW).
Ruijie# **show ip ref mpls forwarding-table vfi** [*vpls_name*] [**mac-address-table** [*H.H.H*] [**statistics**]	Displays all MAC addresses under the VPLS instance including static MAC addresses learned from PWs and ACs and configured by the user, and statistics of VPLS forwarding.
Ruijie# **show mpls forwarding-table**	Displays PW-related ILM and FTN forwarding table entries.
Ruijie# **show mpls ldp neighbor**	Displays all LDP neighbor information.
Ruijie# **show mpls ldp vc**	Displays all LDP VC information.

Configuring Kompella VPLS

Configuring the Public Network Tunnel Between PEs

You must set up an LSP tunnel must be set up on the public network to provide VPLS services. To run MPLS on the backbone network, you must enable LDP on Ps and PEs to establish a public network tunnel. This means that you have to configure LDP for MPLS devices and enable MPLS forwarding on each interface. The configuration procedure is as follows:

Command	Function
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie# **mpls ip**	Enables MPLS forwarding globally. Caution This command is not applicable to switch chip forwarding.
Ruijie(config)# **mpls router ldp**	Enables LDP and enters LDP configuration mode.

Command	Function
Ruijie(config-mpls-router)# **ldp router-id interface loopback** *id* [**force**]	Configures LDP's router ID, which is usually the IP address of the loopback interface.
Ruijie(config-mpls-router)# **exit**	Exits the LDP configuration mode.
Ruijie(config)# **interface** *type ID*	Enters public network interface configuration mode.
Ruijie(config-if-*type ID*)# **label-switching**	Enables the interface's MPLS forwarding function.
Ruijie(config-if-*type ID*)# **mpls ip**	Enables the interface's LDP function and MPLS forwarding function.
Ruijie(config-if-*type ID*)# **ip ref**	Enables the fast forwarding function for routers.
Ruijie(config-if-type ID)# **end**	Returns to privileged mode.
Ruijie# **copy running-config startup-config**	Saves configuration.

# Configure the public network tunnel between PEs.

```
Ruijie# configure terminal
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0 force
Ruijie(config-mpls-router)# exit
Ruijie(config)# interface gigabitethernet 1/1
Ruijie(config-if-GigabitEthernet 1/1)# label-switching
Ruijie(config-if-GigabitEthernet 1/1)# mpls ip
Ruijie(config-if-GigabitEthernet 1/1)# end
Ruijie# copy running-config startup-config
```

Caution

LDP is a topology-driven protocol. To ensure normal working of the LDP, enable IPv4 routing protocols and ensure their normal operations.

Configuring L2VPN VPLS Address Family

Kompella VPLS uses MP-BGP4 as signaling and auto-discovery mechanism. PE devices in one VPLS instance must exchange VPLS information through the L2VPN address family. By default, the L2VPN VPLS address family is not supported. The procedure for enabling the L2VPN address family is as follows:

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **router bgp** *asn-num*	Creates the BGP and enters BGP configuration mode.
Ruijie(config)# **neighbor** *peer-address* **remote-as** *asn-number*	Sets up an IBGP session.
Ruijie(config)# **neighbor** *peer-adddress* **update-source** **i***nterface-name*	Enables the IBGP session to use the address of the loopback interface as the session's source address.
Ruiije(config-router)# **address-family l2vpn vpls**	Enters the L2VPN VPLS address family.

Command	Function
Ruijie(config-router-af)# **neighbor** *ip-address* **activate**	Activates switching of VPLS information on the BGP session.
Ruijie(config-router-af)# **neighbor** { *ip-address* | *peer-group-name* } **send-community** [**both | standard | extended**]	Specifies the extended community attribute to be sent to BGP neighbors.
Ruijie(config-router-af)# **end**	Returns from address family configuration mode to privileged mode.
Ruijie# **show bgp l2vpn vpls** { **all** [**id:offset | neighbor ip-address | summary**] }	Displays L2VPN VPLS address family information.

# Configure the L2VPN address family and enable VPLS information switching.

```
Ruijie# configure terminal
Ruijie(config)# router bgp 100
Ruijie(config-router)# neighbor 10.10.10.1 remote-as 100
Ruijie(config-router)# neighbor 10.10.10.1 update-source loopback 0
Ruiije(config-router)# address-family l2vpn vpls
Ruijie(config-router-af)# neighbor 10.10.10.1 activate
Ruijie(config-router-af)# neighbor 10.10.10.1 send-community extended
Ruijie(config-router-af)# end
Ruijie# show bgp l2vpn vpls all
```

Configuring Kompella VPLS instance

The **l2 vfi** command can be used to create Kompella VPLS instances or enter Kompella VPLS configuration mode. The **no l2 vfi** command can be used to delete VPLS instances. The unique local VPLS instance name and the unique local device VPN ID must be specified when the instance is being created. The auto-discovery function must be enabled for the specified VPLS instance. Each VPLS name corresponds to a VPN ID. In auto-discovery mode, BGP is the default signaling.

Command	Function
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **l2 vfi** *vpls-name* **vpnid** *vpn-id* **autodiscovery**	Creates the Kompella VPLS instance and enters VPLS configuration mode.
Ruijie(config-vfi)# **signal bgp**	Configures the VPLS instance signaling, which is BGP by default.
Ruijie(config-vfi)# **rd** *rd_value*	Configures the RD value. Configure the RD first.
Ruijie(config-vfi)# **encapsulation mpls** [**ethernet | ethernetvlan**]	Specifies the encapsulation mode of Kompella VPLS PWs, which is Ethernet (that is, raw mode) by default. If the sub-interface VPLS access mode is adopted, we recommend the **ethernetvlan** encapsulation mode (that is, tag mode).
Ruijie(config-vfi)# **route-target** { **import | export | both** } *rt_value*	Configures RTs. Multiple RTs can be configured.

Command	Function
Ruijie(config-vfi)# **site-id** *id* [**site-range** *size*]	Configures VE IDs of VPLS sites and the site range. If the site range is not configured, the default value 16 will be adopted.
Ruijie(config-vfi-site)# **xconnect interface** *interface-id*	Configures the interface bound by the site.
Ruijie(config-vfi-site)# **end**	Exits site configuration mode.
Ruijie# **copy running-config startup-config**	Saves configuration.

# Configure a Kompella VPLS instance.

```
Ruijie#configure terminal
Ruijie(config)# l2 vfi vpls-1 vpnid 1 autodiscovery
Ruijie(config-vfi)# signal bgp
Ruijie(config-vfi)# encapsulation mpls ethernet
Ruijie(config-vfi)# rd 100:1
Ruijie(config-vfi)# route-target both 4500:2
Ruijie(config-vfi)# site-id 1
Ruijie(config-vfi-site)# xconnect interface gigabitethernet 1/1
Ruijie(config-vfi-site)#end
Ruijie# copy running-config startup-config
```

⚠ Caution

The **autodiscovery** keyword must be specified after the **l2 vfi** command to create a Kompella VPLS instance.

The auto-discovery between PEs in a VPLS instance is based on MP-BGP. To configure the Kompella VPLS, specify the VPLS PW signaling protocol as BGP.

VPLS instances of one VPN must be configured with the same ID to facilitate management.

PEs on one VPLS must be configured with the same VPLS PW encapsulation mode. Otherwise, VPLS packets cannot be forwarded. Assuming that a VPLS-X exists, if the VPLS-X encapsulation mode on PE1 is **ethernet** (that is, raw mode), the VPLS encapsulation mode on PE2 must not be **ethernetvlan** (that is, tag mode).

☑ For switches, DHCP packets cannot be transmitted transparently after **ip dhcp snooping** is enabled globally.

Configuring User Access VPLS

Configuring VPLS Access Mode for Switches

A created VPLS instance takes effect only after the user configured with the VPLS instance is connected to the link. Currently, there are the following three ways for switch users to access VPLS services:

☑ For switches, a VPLS instance can bind only one SVI interface because VPLS is supported only by the SVI interface.

⚠ Caution

When the port protection mode is enabled on the AC-end member port of L2VPN, the port protection mode does not take effect on the member port if the corresponding member port is not a Trunk interface.

- VLAN access interface access

This mode applies when user packets transmitted on the VPLS AC are not encapsulated by 802.1Q (that is, the packets do not carry VLAN tags).

Command	**Function**
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **interface** *type ID*	Enters the specified physical interface.
Ruijie(config-if-*type ID*)# **switchport mode access**	Enables the interface to work in access mode.
Ruijie(config-if-*type ID*)# **switchport access** *vlan-id*	Sets the interface as a VLAN member port.
Ruijie(config-if-*type ID*)# **exit**	Exits interface configuration mode.
Ruijie(config)# **interface vlan** *vlan-id*	Enters VLAN interface mode.
Ruijie(config-if-*type ID*)# **exit**	Returns to global configuration mode.
Ruijie(config)# **l2 vfi** *vfi_name* **vpnid** *vpn_id* **autodiscovery**	Enters VFI configuration mode.
Ruijie(config-vfi)# **site-id i***d* [**site-range** *size*]	Configures site information of the VFI.
Ruijie(config-vfi-site)# **xconnect interface vlan** *id*	Configures the AC interface bound by the VPLS site.
Ruijie(config-vfi-site)# **end**	Returns to privileged configuration mode.
Ruijie# **copy running-config startup-config**	Saves configuration.

Caution

For the access interface access mode, we recommend setting the VPLS PW encapsulation mode to **ethernet** (that is, raw mode).

VLAN trunk interface access

This mode applies when multiple VPLS services are transmitted on a physical AC. Each VLAN corresponds to a VPLS instance. PE devices can match packets with VPLS instances according to VLAN tags carried by the user packets to provide the multiplexing of access interfaces.

Command	**Function**
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **interface** *type ID*	Enters the specified physical interface.
Ruijie(config-if-*type ID*)# **switchport mode trunk**	Enables the interface to work in trunk mode.
Ruijie(config-if-*type ID)#* **switchport trunk allow vlan add** *vlan-list*	Sets the VLAN flows allowed to be transmitted on the trunk link.
Ruijie(config-if-*type ID*)# **exit**	Exits interface configuration mode.
Ruijie(config)# **interface vlan** *vlan-id*	Enters VLAN interface mode.
Ruijie(config-if-*type ID*)# **exit**	Returns to global configuration mode.
Ruijie(config)# **l2 vfi** *vfi_name* **vpnid** *vpn_id* **autodiscovery**	Enters VFI configuration mode.
Ruijie(config-vfi)# **site-id** *id* [**site-range** *size*]	Configures site information of the VFI.
Ruijie(config-vfi-site)# **xconnect interface vlan** *id*	Configures the AC interface bound by the VPLS site.
Ruijie(config-vfi-site)# **end**	Returns to privileged mode.
Ruijie# **copy running-config startup-config**	Saves configuration.

Caution

For the Trunk interface access mode, we recommend setting the VPLS PW encapsulation mode to **ethernetvlan** (that is, tag mode).

The L2 VPN service cannot be bound to the Native VLAN of the trunk interface.

- VLAN tunnel interface access

This mode applies when user service packets transmitted on ACs carry private VLAN tags if the user is connected to the VPLS service. In this mode, all packets received by PEs from the interface are forwarded without being processed. This mode requires the VLAN member port between PEs and CEs to work in tunnel mode.

Command	Function
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **interface** *type ID*	Enters the specified physical interface.
Ruijie(config-if-*type ID*)# **switchport access** *vlan-id*	Sets the interface as a VLAN member port.
Ruijie(config-if-*type ID*)# **switchport mode dot1q-tunnel**	Enable the interface to work in tunnel mode.
Ruijie(config-if-*type ID*)# **exit**	Exits interface configuration mode.
Ruijie(config)# **interface vlan** *vlan-id*	Enters VLAN interface mode.
Ruijie(config-if-*type ID*)# **exit**	Returns to global configuration mode.
Ruijie(config)# **l2 vfi** *vfi_name* **vpnid** *vpn_id* **autodiscovery**	Enters VFI configuration mode.
Ruijie(config-vfi)# **site-id** *id* [**site-range** *size*]	Configures site information of the VFI.
Ruijie(config-vfi-site)# **xconnect interface vlan** *id*	Configures the AC interface bound by the VPLS site.
Ruijie(config-vfi-site)# **end**	Returns to privileged mode.
Ruijie# **copy running-config startup-config**	Saves configuration.

Caution

For the VLAN tunnel interface access mode, we recommend setting the VPLS PW encapsulation mode to **ethernet** (that is, raw mode).

For the VLAN tunnel interface access mode, only the basic QinQ is supported.

Configuring VPLS Access Mode for Routers

A created VPLS instance takes effect only after the user configured with the VPLS instance is connected with the link.

There are the following two ways to access VPLS services:

- Ethernet interface access

This mode applies user service packets transmitted on ACs carry private VLAN tags or do not carry VLAN tags in the case of VPLS service access. In this mode, all packets received by PEs from the interface are forwarded according to the destination MAC address and the private tags are considered part of the data. In such case, PEs provide port-based VPLS services. The address learning mode of the VPLS instance of the bound port is the free mode. MAC address overlapping in the user VLAN is not supported.

Command	Function
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **interface** *type ID*	Enters the specified physical interface.

Command	**Function**
Ruijie(config-if-*type ID*)# **ip ref**	Enables the fast forwarding function for routers.
Ruijie(config-if-*type ID*)# **end**	Returns to privileged mode.
Ruijie(config-if-*type ID*)# **exit**	Returns to global configuration mode.
Ruijie(config)# **l2 vfi** *vfi_name* **vpnid** *vpn_id* **autodiscovery**	Enters VFI configuration mode.
Ruijie(config-vfi)# **site-id** *id* [**site-range** *size*]	Configures site information of the VFI.
Ruijie(config-vfi-site)#**xconnect interface** *interface-id*	Configures the AC interface bound by the VPLS site.
Ruijie(config-vfi-site)#**end**	Returns to privileged mode.
Ruijie# **copy running-config startup-config**	Saves configuration.

# Configure Ethernet interface VPLS access.

```
Ruijie# configure terminal
Ruijie(config)# interface gigabitethernet 1/2
```

# Enable the fast forwarding function of the interface for routers.

```
Ruijie(config-if-GigabitEthernet 1/2)# ip ref
Ruijie(config-if-GigabitEthernet 1/2)# exit
Ruijie(config)# l2 vfi vfiA vpnid 1 autodiscovery
Ruijie(config-vfi)# rd 2:2
Ruijie(config-vfi)# route-target both 2:2
Ruijie(config-vfi)# encapsulation mpls ethernet
Ruijie(config-vfi)# site-id 1
Ruijie(config-vfi-site)# xconnect interface gigabitethernet 1/2
Ruijie(config-vfi-site)#end
Ruijie# copy running-config startup-config
```

Caution

For the Ethernet access mode, we recommend using the **encapsulation mpls ethernet** command to specify the Kompella VPLS PW encapsulation mode as **ethernet** (that is, raw mode). If the packets of imported PEs carry VLAN tags, the tags will be transmitted transparently as private tags.

☑ For routers, a VPLS instance can be configured with mixed access modes. For example, PEs in a VPLS instance are configured with both Ethernet and sub-interface access modes. In such case, we recommend setting the VPLS PW encapsulation mode on the PEs to **ethernetvlan** (that is, tag mode).

- Sub-interface access

PE devices can access a VPLS instance through sub-interfaces.

Command	**Function**
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **interface** *type ID*	Enters the specified physical interface.
Ruijie(config-if-*type ID*)# **ip ref**	For routers, the fast forwarding function must be enabled on the master interface of the sub-interface.
Ruijie(config-if-*type ID*)# **end**	Returns to privileged mode.
Ruijie(config-if-*type ID*)# e**xit**	Returns to global configuration mode.

Ruijie(config)# **l2 vfi** *vfi_name* **vpnid** *vpn_id* **autodiscovery**	Enters VFI configuration mode.
Ruijie(config-vfi)# **site-id** *id* [**site-range** *size*]	Configures site information of the VFI.
Ruijie(config-vfi-site)#**xconnect interface** *interface-id*	Configures the AC interface bound by the VPLS site.
Ruijie(config-vfi-site)#**end**	Returns to privileged mode.
Ruijie# **copy running-config startup-config**	Saves configuration.

# Configure sub-interface VPLS access.

```
Ruijie# configure terminal
```

# Enable the fast forwarding function of the sub-interface for routers.

```
Ruijie(config)# interface gigabitethernet 1/2
Ruijie(config-if-GigabitEthernet 1/2)# ip ref
Ruijie(config-if-GigabitEthernet 1/2)# exit
Ruijie(config)# interface gigabitethernet 1/2.10
Ruijie(config-if-GigabitEthernet 1/2.10)# encapsulation dot1Q 10
Ruijie(config-if-GigabitEthernet 1/2.10)#exit
Ruijie(config)# l2 vfi vfiA vpnid 1 autodiscovery
Ruijie(config-vfi)# rd 2:2
Ruijie(config-vfi)# route-target both 2:2
Ruijie(config-vfi)# encapsulation mpls ethernetvlan
Ruijie(config-vfi)# site-id 1
Ruijie(config-vfi-site)# xconnect interface gigabitethernet 1/2.10
Ruijie(config-vfi-site)#end
Ruijie# copy running-config startup-config
```

Caution

For the sub-interface access mode, we recommend using the **encapsulation mpls ethernetvlan** command to specify the Kompella VPLS PW encapsulation mode as **ethernetvlan** (that is, tag mode).

Configuring Kompella VPLS's Compatibility (Optional)

By default, the PW's mtu value provided by L2VPN is 1500 bytes. If the same PW's mtu values on two PEs are different, PW connection cannot be set up between the two PEs. Some manufacturers' devices do not support configuring mtu in L2VPN instances. When such devices perform Kompella communication with devices of other manufacturers, the **ingore match l2-extcommunity** command can be used to ignore received mtu and matching detection of **Control Flag**, ensuring that the VC link is UP.

Command	Function
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **l2 vfi** *vpls-name* **vpnid** *vpn-id* **autodiscovery**	Creates the Kompella VPLS instance and enters VPLS configuration mode.
Ruijie(config-vfi)# **ingore match l2-extcommunity**	Configures ignoring detection of L2VPN expanded community attribute members.
Ruijie# **copy running-config startup-config**	Saves configuration.

Caution This command takes effect only on Kompella L2VPN.

Verifying Kompella VPLS Configuration

Command	Function
show bgp l2vpn vpls all	Displays NLRI information about all VPWS instances.
show bgp l2vpn vpls all connections	Displays all connection information of Kompella VPLS.
show mpls l2transport vc [*vc_id* [*ip-address*]] | [**interface** *interface_name*] [**detail**]	Displays VC information established by Kompella VPLS.
show mpls vfi [*name*]	Displays VPLS instance information.
Ruijie# **show ip ref mpls forwarding-table vfi** [*vpls_name*] [**mac-address-table** [*H.H.H*] **| statistics**]	Displays all MAC addresses under the VPLS instance including static MAC addresses learned from PWs and ACs and configured by the user, and statistics of VPLS forwarding.
Ruijie# **show mpls forwarding-table**	Displays PW-related ILM and FTN forwarding table entries.

Configuring Other VPLS Parameters

Configuring VPLS Instance Descriptors (Optional)

You can configure the descriptive information of each VPLS instance.

Command	Function
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **l2 vfi** *vpls-name*	Enters VPLS configuration mode.
Ruijie(config-vfi)# **description** *vpls-description*	(Optional) Configures descriptive information of the VPLS.
Ruijie(config-vfi)# **end**	Returns to privileged mode.
Ruijie# **copy running-config startup-config**	Saves configuration.

Configuring mtu of VPLS Instance (Optional)

You can configure each VPLS instance's mtu value, which is **1500** by default. The mtu value of VPLS indicates the length of the packet that can be transmitted by the PW, or the length of the user's layer-2 packet plus the length of the PW-encapsulated packet. By default, if the PW does not enable the control word, assuming that two labels are encapsulated, the length of an Ethernet packet that can be transmitted is 1492 bytes, of which 8 bytes are encapsulated by the PW (2 labels).

Command	Function
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **l2 vfi** *vpls-name*	Enters VPLS configuration mode.
Ruijie(config-vfi)# **mtu** *mtu*	(Optional) Configures the mtu value of the VPLS.
Ruijie(config-vfi)# **end**	Returns to privileged mode.
Ruijie# **copy running-config startup-config**	Saves configuration.

Caution

The mtu values of one VPLS instance on different PEs must be the same. Otherwise, the signaling protocol cannot establish PWs.

Currently, transmission of VPLS fragments is not supported. If the PW signaling protocol negotiation's mtu is modified, the mtu of the user access service interface must be adjusted (generally adjusted to the PW mtu length minus the encapsulated length); the PW's public-network-end output interface's mtu, MPLS mtu and PW mtu must be the same to ensure proper forwarding. The **mtu** command can be used on an interface to modify the interface's mtu. Use the **mpls mtu** command to modify MPLS mtu of the interface.

Configuring VPLS Transparent Transmission Bridge Protocol Control Packet (Optional)

You can control whether the VPLS interface transparently transmits bridge protocol control packets to meet application needs.

Command	Function
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **interface** *type ID*	Enters specified interface mode.
Ruijie(config-if-*type ID*)# **l2 vfi tunnel-protocol stp**	Enables the interface to transparently transmit STP packets.
Ruijie(config-if-*type ID*)# **end**	Returns to privileged mode.
Ruijie# **copy running-config startup-config**	Saves configuration.

Caution

Generally, BPDU packets do not carry VLAN Tag. If CEs access PEs through a trunk interface or sub-interface and the BPDU transparent transmission function is enabled on the access interface, the BPDU packets sent by CEs must carry corresponding VLAN tags so that the BPDU packets can be identified by the corresponding VPLS instances and transmitted transparently in the VPLS instances.

Configuring MAC Address Aging Time (Optional)

You can configure each VPLS instance's address aging time, which is 300 seconds by default.

Command	Function
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **l2 vfi** *vpls-name*	Enters VPLS configuration mode.
Ruijie(config-vfi)# **mac-address aging-time** *interval*	(Optional) Configures the address aging time of the VPLS.
Ruijie(config-vfi)# **end**	Returns to privileged mode.
Ruijie# **copy running-config startup-config**	Saves configuration.

Caution

If the aging time of MAC addresses is changed, all the MAC addresses will be aged according to the new aging time. If the aging time is set to *m*, MACs that have not performed communication will be aged after *m*. Configuring MAC Address Capacity Volume Limit (Optional)

You can configure each VPLS instance's MAC address capacity limit and behavior to be performed when the limit is exceeded.

Command	Function
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **l2 vfi** *vpls-name*	Enters VPLS configuration mode.
Ruijie(config-vfi)# **mac-limit** { **action** { **discard | forward** } **|** **alarm** { **disable | enable** } **| maximum** *count*	(Optional) Configures each VPLS instance's MAC address capacity limit and behavior to be performed when the limit is exceeded.
Ruijie(config-vfi)# **end**	Returns to privileged mode.
Ruijie# **copy running-config startup-config**	Saves configuration.

Caution

By default, the message warning of exceeding the capacity is disabled. If it is enabled, the Log message will be displayed when the MAC capacity of the VPLS instance is exceeded for the first time. The Log message will be displayed again when the VPLS MAC capacity drops to below the limit.

Configuring Static MAC Addresses (Optional)

You can configure static MAC addresses for VPLS instances. When conflicts exist in static MAC addresses and dynamically learned ones, dynamically learned MAC addresses are overwritten.

Command	Function
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie(config)# **mpls static vfi** *vpls-name* **mac-address** *H.H.H* **neighbor** *ip-address*	Configures MAC addresses for a VPLS instance.
Ruijie(config)# **end**	Returns to the privileged mode.
Ruijie# **copy running-config startup-config**	Saves the configuration.

Caution

Static MAC addresses configured through configuration of relate PWs of neighbor addresses can work properly only in the following scenario: The L2VPN instance has only a PW for the neighbor configured. If the L2VPN instance is configured with multiple PWs for the neighbor, the configured static MAC address will bind with a PW randomly, leading to incorrect forwarding.

Clearing Dynamic MAC Addresses (Optional)

A VPLS instance can dynamically learn MAC addresses from ACs and PWs. You can clear the MAC addresses learned dynamically by VPLS instances, including the local and remote VPLS instances.

Command	Function
Ruijie# **config terminal**	Enters global configuration mode.
Ruijie# **clear l2 vfi** *vpls-name* mac-**address local**	Clears the MAC addresses learned dynamically by the local (local PE) VPLS instance.
Ruijie# **clear l2 vfi** *vpls-name* mac-address **remote**	Clears the MAC addresses learned dynamically by the remote (another PE) VPLS instance.

Caution

Clearing MAC addresses of remote PEs is effective for Martini VPLS only.

Applying Tunnel Policy (Optional)

Configuring Tunnel Policy

For details, see Section 3.2.2.6.

Applying Tunnel Policy to Martini VPLS

Command	Function
Ruijie# **config terminal**	Enters the global configuration mode.
Ruijie(config)# **l2 vfi** *vpls-name* **vpnid** *vpn-id*	Creates a VPLS instance and enters the VPLS configuration mode.
Ruijie(config-vfi)# **neighbor** *ip-address* **encapsulation mpls** [**vc-id** *vc-id*] [**hub-vc** | **spoke-vc**] [**ethernet** | **ethernetvlan**] [**tnl-policy** *policy-name*]	Configures a VPLS peer and applies the tunnel policy.

#Example: Apply the tunnel policy to the Martini VPLS.

```
Ruijie# configure terminal
Ruijie(config)# l2 vfi vpls-1 vpnid 1
Ruijie(config-vfi)# neighbor 2.2.2.2 encapsulation mpls vc-id 1 tnl-policy pol
Ruijie(config-vfi)# exit
```

Applying Tunnel Policy to Kompella VPLS

Command	Function
Ruijie# **config terminal**	Enters the global configuration mode.
Ruijie(config)# **l2 vfi** *vpls-name* **vpnid** *vpn-id* **autodiscovery**	Creates a Kompella VPLS instance and enters the VPLS configuration mode.
Ruijie(config-vfi)# **neighbor** *peer-address* **tnl-policy** *policy-name*	Applies the tunnel policy to the VFI.

#Example: Apply the tunnel policy to the Kompella VPLS.

```
Ruijie# configure terminal
Ruijie(config)# l2 vfi vpls-2 vpnid 2 autodiscovery
Ruijie(config-vfi)# rd 2:2
Ruijie(config-vfi)# neighbor 2.2.2.2 tnl-policy pol
Ruijie(config-vfi)# exit
```

Configuring PW Class (Optional)

Configure the PW class instance to be referenced by the VPLS. Use the **pw-class** command to create a PW template instance and enter the PW template instance. Use the **no** form of this command to delete the PW template instance. You must specify a locally unique name for the PW template instance when you create it.

Command	Function
Ruijie# **configure terminal**	Enters the global configuration mode.
Ruijie(config)#**pw-class** *pwc1*	Creates a PW class instance and enters the PW class configuration mode.
Ruijie(config-pw-class)#**backup neighbor** *ip-address* [**vc-id** *vc_id*] [**manual**]	Configures the IP address and VC identifier of the remote neighbor of the backup PW. If no VC identifier is configured, reuse the VC identifier of the primary PW. The other attributes of the standby PW also use the attributes of the primary PW.
Ruijie(config-pw-class)# **vccv bfd** [**primary** | **backup**] **enable**	Enables the PW connectivity check. The PW connectivity check is enabled on the primary and backup PWs by default. The PW connectivity check in BFD mode is supported.
Ruijie(config-pw-class)# **fast-reroute** [**delay** *delay-time* | **immediately** | **never**]	Configures the switchback policy for switching from the backup PW back to primary PW. The switchback to the primary PW is delayed for 300s by default.
Ruijie(config-pw-class)# **end**	Returns to the privileged EXEC mode.
Ruijie# **copy running-config startup-config**	Saves the configuration.

Configuring VPLS Referencing PW Class (Optional)

Run the following commands to enable the VPLS to reference the PW class instance.

Command	Function
Ruijie# **config terminal**	Enters the global configuration mode.
Ruijie(config)# **l2 vfi** *vpls-name* **vpnid** *vpn-id*	Creates a VPLS instance and enters the VPLS configuration mode.
Ruijie(config-vfi)# **neighbor** *ip-address* **encapsulation mpls** [**vc-id** *vc-id*] [**hub-vc** | **spoke-vc**] [**ethernet** | **ethernetvlan**] [**manual**] [**tnl-policy** *policy-name*] [**pw-class** *class-name*]	Configures a VPLS peer and reference the PW class instance.
Ruijie(config-vfi)# **end**	Returns to the privileged EXEC mode.
Ruijie# **copy running-config startup-config**	Saves the configuration.

Caution

You must create a PW class instance before it can be referenced.

#Example: Enable the Martini VPLS to reference the PW class instance.

```
Ruijie# configure terminal
Ruijie(config)# l2 vfi vfi_frr vpnid 1
Ruijie(config-vfi)# neighbor 2.2.2.2 encapsulation mpls pw-class pwc1
Ruijie(config-vfi)# exit
```

Configuring MAC Address Learning of VPLS (Optional)

The VPLS can dynamically learn the MAC addresses of the AC and the PW. You can control the MAC address learning function on the VPLS, including enabling or disabling learning of the AC MAC address and learning of the PW MAC address.

Command	Function
Ruijie# **config terminal**	Enters the global configuration mode.
Ruijie(config)# **l2 vfi** *name* **vpnid** *vpn-id*	Creates a VPLS instance and enters the VPLS configuration mode.
Ruijie(config-vfi)# **mac-learning** [**ac|pw**] **disable**	Controls MAC address learning. The MAC addresses of the AC and the PW are learned concurrently by default.
Ruijie(config-vfi)# **end**	Returns to the privileged EXEC mode.
Ruijie# **copy running-config startup-config**	Saves the configuration.

Caution This command is available to both the Martini VPLS and the Kompella VPLS.

Configuring MAC Address Withdrawal of VPLS (Optional)

You can withdraw the remote MAC address dynamically learned by the VPLS and forward the MAC address withdrawal message to other PE devices.

Command	Function
Ruijie# **config terminal**	Enters the global configuration mode.
Ruijie(config)# **l2 vfi** *name* **vpnid** *vpn-id*	Creates a VPLS instance and enters the VPLS configuration mode.
Ruijie(config-vfi)# **mac-withdraw enable**	Enables the MAC address withdrawal message sending function. If this function is disabled, the local spoke PW is invalid or the AC is invalid. In this case, no MAC address withdrawal message is sent to the remote peer. Upon receiving the MAC address withdrawal message from the hub PW, the PE device withdraws the MAC address and sends the MAC address withdrawal message other PEs.
Ruijie(config-vfi)# **end**	Returns to the privileged EXEC mode.
Ruijie# **copy running-config startup-config**	Saves the configuration.

Caution Withdrawing the MAC address of the remote PE is available to the Martini VPLS only.

Configuring Action on Unknown Unicast Packets Received by VPLS Instance (Optional)

Run the following commands to configure the forwarding action on the unknown unicast packets received by the VPLS instance.

Command	Function
Ruijie# **config terminal**	Enters the global configuration mode.
Ruijie(config)# **l2 vfi** *name* **vpnid** *vpn-id*	Creates a VPLS instance and enters the VPLS configuration mode.
Ruijie(config-vfi)# **unknown-frame unicast** [**ac-ac |ac-pw** | **pw-ac**] **discard**	Configures the forwarding action on the unknown unicast and broadcast packets.
Ruijie(config-vfi)# **end**	Returns to the privileged EXEC mode.
Ruijie# **copy running-config startup-config**	Saves the configuration.

Maintaining PW Connectivity of VPLS

Configuration Prerequisites

When using VCCV to detect the PW connectivity of the VPLS, you must ensure that the PWE3 network is already correctly configured.

Checking PW Connectivity

Command	Function
Ruijie#**ping mpls pseudowire** { *ip-address vc-id* [**segment** *segment-number*] | **kompella** *vfi_name local_site_id remote_site_id* **rd** *remote_rd* } { **label-alert** | **ttl-expiry** } [**repeat** *repeat*] [**timeout** *timeout*] [**size** *size*] [**interval** *mseconds*] [**source** *ip-address*] [**destination** *ip-address*] [**pad** *pattern*] [**reply mode** { **ipv4** | **router-alert | control-channel** }] [**verbose**]	Checks the PW connectivity.

Example: Check the connectivity of the PW whose peer PE IPv4 address is 10.10.10.10 and PW ID is 100.

```
Ruijie#ping mpls pseudowire 10.10.10.3 100 label-alert
Sending 5, 112-byte MPLS Echoes on Pseudowire, peer address 10.10.10.3, VC ID 100,
     timeout is 2 seconds, send interval is 0 msec:

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0
```

```
Press Ctrl+C to break.
!!!!!
Success rate is 100 percent(5/5),round-trip min/avg/max=20/36/60 ms
```

Locating Data Plane Fault of PW

Command	Function
Ruijie#**traceroute mpls pseudowire** { *ip-address vc-id* {**segment** *segment-number*} | **kompella** *vfi_name local_site_id remote_site_id* **rd** *remote_rd* } { **label-alert** | **ttl-expiry** } [**repeat** *repeat*] [**timeout** *timeout*] [**size** *size*] [**interval** *mseconds*] [**source** *ip-address*] [**destination** *ip-address*] [**pad** *pattern*] [**reply mode** { **ipv4** | **router-alert** }] [**verbose**]	Tracks the PW path to locate the connectivity fault.

Example: Track the path of the PW whose VFI name is vpn-1, local VE ID is 1, remote VE ID is 2, and remote RD is 200:1 to locate the fault.

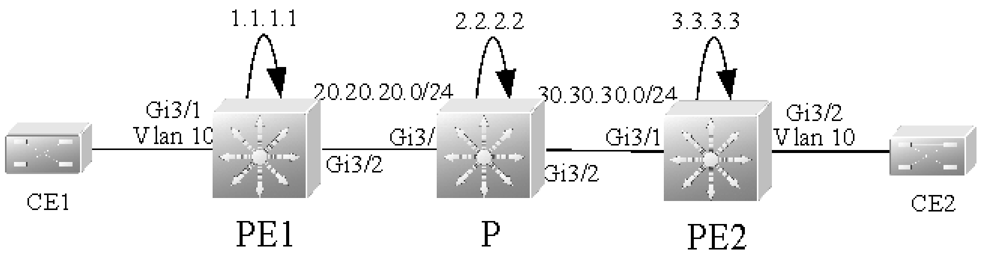
```
Ruijie#traceroute mpls pseudowire kompella vpn-1 1 2 rd 200:1 label-alert
Traceing Pseudowire Label Switched Path, VFI name vpn-1, local site id 1, remote site id 2,
timeout is 2 seconds.
Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Press Ctrl+C to break.
  0 192.168.1.10   MRU 1500 [Labels: router-alert/1538 Exp: 0/0]
! 1 192.168.1.11   20 ms
```

Typical Examples of Martini VPLS Configuration for Switches

Basic VPLS

As shown in the following figure, CE1 and CE2 access the same VPLS network through PE1 and PE2. PE1, P, and PE2 form a public MPLS network that provides VPLS services. On PE1 and PE2, bind VLAN 10 to the VPLS instance.

Figure 71



The configuration procedure is as follows:

Configuring PE1:

# Configure the loopback interface.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 1.1.1.1 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 1.1.1.1 0.0.0.0 area 0
Ruijie(config-router)# network 20.20.20.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# neighbor 3.3.3.3
Ruijie(config-mpls-router)# exit
```

# Enable LDP and MPLS on the public network interface.

```
Ruijie(config)# interface gigabitEthernet 3/2
Ruijie(config-if-Gigabitethernet 3/2)# no switchport
Ruijie(config-if-Gigabitethernet 3/2)# ip address 20.20.20.1 255.255.255.0
Ruijie(config-if-Gigabitethernet 3/2)# mpls ip
Ruijie(config-if-Gigabitethernet 3/2)# label-switching
Ruijie(config-if-Gigabitethernet 3/2)# exit
```

# Configure a VPLS instance and specify the peer PE.

```
Ruijie(config)# l2 vfi vfi_a vpnid 1
Ruijie(config-vpls)# neighbor 3.3.3.3 encapsulation mpls
Ruijie(config-vpls)# exit
```

# Bind the VLAN interface to the VPLS instance.

```
Ruijie(config)# vlan 10
Ruijie(config-vlan)# exit
Ruijie(config)# interface vlan 10
Ruijie(config-if-Vlan 10)# xconnect vfi vfi_a
Ruijie(config-if-Vlan 10)# exit
```

# Configure the interface between PEs and CEs.

```
Ruijie(config)# interface gigabitEthernet 3/1
Ruijie(config-if-GigabitEthernet 3/1)# switchport access vlan 10
Ruijie(config-if-GigabitEthernet 3/1)# exit
```

Configuring P:

# Configure the loopback interface.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 2.2.2.2 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 2.2.2.2 0.0.0.0 area 0
Ruijie(config-router)# network 20.20.20.0 0.0.0.255 area 0
Ruijie(config-router)# network 30.30.30.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# exit
```

# Enable LDP and MPLS on an interface.

```
Ruijie(config)# interface gigabitEthernet 3/1
Ruijie(config-if-Gigabitethernet 3/1)# no switchport
Ruijie(config-if-Gigabitethernet 3/1)# ip address 20.20.20.2 255.255.255.0
Ruijie(config-if-Gigabitethernet 3/1)# mpls ip
Ruijie(config-if-Gigabitethernet 3/1)# label-switching
Ruijie(config-if-Gigabitethernet 3/1)# exit
Ruijie(config)# interface gigabitEthernet 3/2
Ruijie(config-if-Gigabitethernet 3/2)# no switchport
Ruijie(config-if-Gigabitethernet 3/2)# ip address 30.30.30.1 255.255.255.0
Ruijie(config-if-Gigabitethernet 3/2)# mpls ip
Ruijie(config-if-Gigabitethernet 3/2)# label-switching
Ruijie(config-if-Gigabitethernet 3/2)# exit
```

Configuring PE2:

# Configure the loopback interface.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 3.3.3.3 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 3.3.3.3 0.0.0.0 area 0
Ruijie(config-router)# network 30.30.30.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# neighbor 1.1.1.1
Ruijie(config-mpls-router)# exit
```

# Enable LDP and MPLS on the public network interface.

```
Ruijie(config)# interface gigabitEthernet 3/1
Ruijie(config-if-Gigabitethernet 3/1)# no switchport
Ruijie(config-if-Gigabitethernet 3/1)# ip address 30.30.30.2 255.255.255.0
Ruijie(config-if-Gigabitethernet 3/1)# mpls ip
Ruijie(config-if-Gigabitethernet 3/1)# label-switching
Ruijie(config-if-Gigabitethernet 3/1)# exit
```

# Configure a VPLS instance and specify the peer PE.

```
Ruijie(config)# l2 vfi vfi_a vpnid 1
Ruijie(config-vpls)# neighbor 1.1.1.1 encapsulation mpls
Ruijie(config-vpls)# exit
```

# Bind the VLAN interface to the VPLS instance.

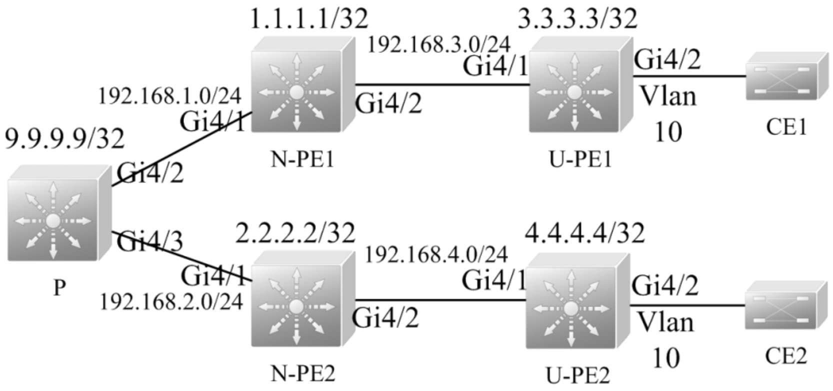
```
Ruijie(config)# vlan 10
Ruijie(config-vlan)# exit
Ruijie(config)# interface vlan 10
Ruijie(config-if-Vlan 10)# xconnect vfi vfi_a
Ruijie(config-if-Vlan 10)# exit
```

# Configure the interface between PEs and CEs.

```
Ruijie(config)# interface gigabitEthernet 3/2
Ruijie(config-if-Gigabitethernet 3/2)#  switchport access vlan 10
Ruijie(config-if-Gigabitethernet 3/2)# exit
```

H-VPLS (PW Access)

Figure 72



As shown in the preceding figure, CE1 and CE2 access the same H-VPLS network through U-PE1 and U-PE2, which are connected to N-PE1 and N-PE2 respectively. The N-PEs and U-PEs are connected through PWs. All U-PEs and N-PEs belong to the MPLS network and work together to provide VPLS services.

When setting up PWs between N-PEs and U-PEs, you must specify the PW of the N-PEs with the U-PEs as a Spoke PW. On the U-PE end, the PW type is not restricted. If U-PEs support VPLS, the PW between U-PEs and N-PEs can be either a Hub PW or a Spoke PW. If the U-PEs do not support VPLS, the VPWS PW can also be configured.

In this example, U-PE1 and N-PE1 are connected through a Hub PW whereas U-PE2 and N-PE2 are connected through a VPWS PW.

- Configuring U-PE1:

# Configure the loopback interface.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 3.3.3.3 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 3.3.3.3 0.0.0.0 area 0
Ruijie(config-router)# network 192.168.3.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# neighbor 1.1.1.1
Ruijie(config-mpls-router)# exit
```

# Enable LDP and MPLS on the public network interface.

```
Ruijie(config)# interface gigabitEthernet 4/1
Ruijie(config-if-GigabitEthernet 4/1)# no switchport
Ruijie(config-if-GigabitEthernet 4/1)# ip address 192.168.3.2 255.255.255.0
Ruijie(config-if-GigabitEthernet 4/1)# mpls ip
Ruijie(config-if-GigabitEthernet 4/1)# label-switching
Ruijie(config-if-GigabitEthernet 4/1)# exit
```

# Configure a VPLS instance and enable a PW to connect the peer N-PE.

```
Ruijie(config)# l2 vfi vfi_a vpnid 1
Ruijie(config-vpls)# neighbor 1.1.1.1 encapsulation mpls
Ruijie(config-vpls)# exit
```

# Bind the VLAN interface to the VPLS instance.

```
Ruijie(config)# vlan 10
Ruijie(config-vlan)# exit
Ruijie(config)# interface vlan 10
Ruijie(config-if-Vlan 10)# xconnect vfi vfi_a
Ruijie(config-if-Vlan 10)# exit
```

# Configure the interface between PEs and CEs.

```
Ruijie(config)# interface gigabitEthernet 4/2
Ruijie(config-if-GigabitEthernet 4/2)# switchport access vlan 10
Ruijie(config-if-GigabitEthernet 4/2)# exit
```

- Configuring N-PE1:

# Configure the loopback interface.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 1.1.1.1 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 1.1.1.1 0.0.0.0 area 0
Ruijie(config-router)# network 192.168.1.0 0.0.0.255 area 0
Ruijie(config-router)# network 192.168.3.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# neighbor 2.2.2.2
Ruijie(config-mpls-router)# neighbor 3.3.3.3
Ruijie(config-mpls-router)# exit
```

# Enable LDP and MPLS on the public network interface.

```
Ruijie(config)# interface gigabitEthernet 4/1
Ruijie(config-if-GigabitEthernet 4/1)# no switchport
Ruijie(config-if-GigabitEthernet 4/1)# ip address 192.168.1.2 255.255.255.0
Ruijie(config-if-GigabitEthernet 4/1)# mpls ip
Ruijie(config-if-GigabitEthernet 4/1)# label-switching
Ruijie(config-if-GigabitEthernet 4/1)# exit
Ruijie(config)# interface gigabitEthernet 4/2
Ruijie(config-if-GigabitEthernet 4/2)# no switchport
Ruijie(config-if-GigabitEthernet 4/2)# ip address 192.168.3.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 4/2)# mpls ip
Ruijie(config-if-GigabitEthernet 4/2)# label-switching
Ruijie(config-if-GigabitEthernet 4/2)# exit
```

# Configure a VPLS instance, a Hub PW to connect another N-PE, and the Spoke PW to connect the peer U-PE.

```
Ruijie(config)# l2 vfi vfi_a vpnid 1
Ruijie(config-vpls)# neighbor 2.2.2.2 encapsulation mpls
Ruijie(config-vpls)# neighbor 3.3.3.3 encapsulation mpls spoke-vc
Ruijie(config-vpls)# exit
```

- Configuring P:

# Configure the loopback interface.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 9.9.9.9 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 9.9.9.9 0.0.0.0 area 0
Ruijie(config-router)# network 192.168.1.0 0.0.0.255 area 0
Ruijie(config-router)# network 192.168.2.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# exit
```

# Enable LDP and MPLS on an interface.

```
Ruijie(config)# interface gigabitEthernet 4/2
Ruijie(config-if-GigabitEthernet 4/2)# no switchport
Ruijie(config-if-GigabitEthernet 4/2)# ip address 192.168.1.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 4/2)# mpls ip
Ruijie(config-if-GigabitEthernet 4/2)# label-switching
Ruijie(config-if-GigabitEthernet 4/2)# exit
Ruijie(config)# interface gigabitEthernet 4/3
Ruijie(config-if-GigabitEthernet 4/3)# no switchport
Ruijie(config-if-GigabitEthernet 4/3)# ip address 192.168.2.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 4/3)# mpls ip
Ruijie(config-if-GigabitEthernet 4/3)# label-switching
Ruijie(config-if-GigabitEthernet 4/3)# exit
```

- Configuring N-PE2:

# Configure the loopback interface.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 2.2.2.2 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 2.2.2.2 0.0.0.0 area 0
Ruijie(config-router)# network 192.168.2.0 0.0.0.255 area 0
Ruijie(config-router)# network 192.168.4.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# neighbor 1.1.1.1
Ruijie(config-mpls-router)# neighbor 4.4.4.4
Ruijie(config-mpls-router)# exit
```

# Enable LDP and MPLS on the public network interface.

```
Ruijie(config)# interface gigabitEthernet 4/1
Ruijie(config-if-GigabitEthernet 4/1)# no switchport
Ruijie(config-if-GigabitEthernet 4/1)# ip address 192.168.2.2 255.255.255.0
Ruijie(config-if-GigabitEthernet 4/1)# mpls ip
Ruijie(config-if-GigabitEthernet 4/1)# label-switching
Ruijie(config-if-GigabitEthernet 4/1)# exit
Ruijie(config)# interface gigabitEthernet 4/2
Ruijie(config-if-GigabitEthernet 4/2)# no switchport
Ruijie(config-if-GigabitEthernet 4/2)# ip address 192.168.4.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 4/2)# mpls ip
Ruijie(config-if-GigabitEthernet 4/2)# label-switching
Ruijie(config-if-GigabitEthernet 4/2)# exit
```

# Configure a VPLS instance, enable a Hub PW to connect another N-PE, and enable the Spoke PW to connect the peer U-PE.

```
Ruijie(config)# l2 vfi vfi_a vpnid 1
Ruijie(config-vpls)# neighbor 1.1.1.1 encapsulation mpls
Ruijie(config-vpls)# neighbor 4.4.4.4 encapsulation mpls  spoke-vc
Ruijie(config-vpls)# exit
```

- Configuring U-PE2:

# Configure the loopback interface.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 4.4.4.4 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 4.4.4.4 0.0.0.0 area 0
Ruijie(config-router)# network 192.168.4.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# neighbor 2.2.2.2
Ruijie(config-mpls-router)# exit
```

# Enable LDP and MPLS on the public network interface.

```
Ruijie(config)# interface gigabitEthernet 4/1
Ruijie(config-if-GigabitEthernet 4/1)# no switchport
Ruijie(config-if-GigabitEthernet 4/1)# ip address 192.168.4.2 255.255.255.0
Ruijie(config-if-GigabitEthernet 4/1)#mpls ip
Ruijie(config-if-GigabitEthernet 4/1)# label-switching
Ruijie(config-if-GigabitEthernet 4/1)#  exit
```

# Configure a VPWS PW for a VLAN interface.

```
Ruijie(config)# vlan 10
Ruijie(config-vlan)# exit
Ruijie(config)# interface vlan 10
Ruijie(config-if-Vlan 10)# xconnect 2.2.2.2 1 encapsulation mpls ethernet
Ruijie(config-if-Vlan 10)# exit
```

# Configure the interface between PEs and CEs.

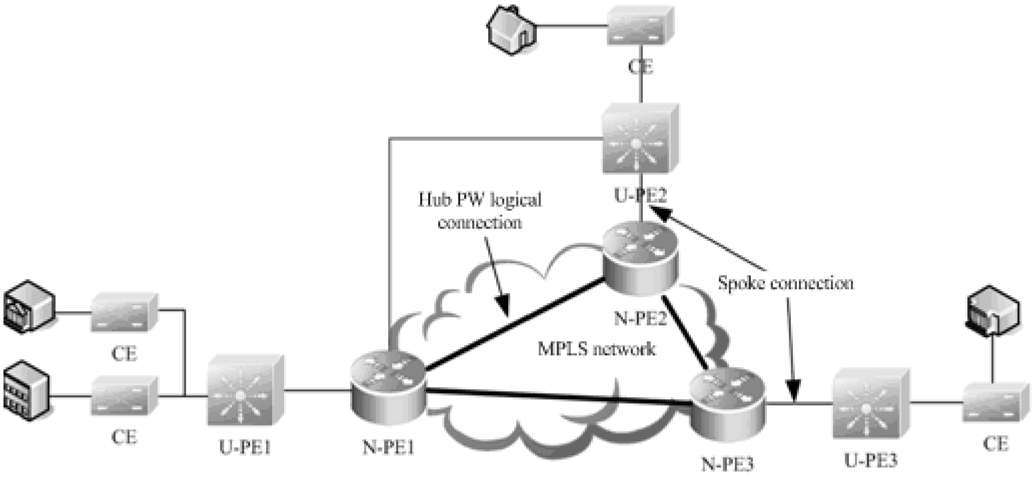
```
Ruijie(config)# interface gigabitEthernet 4/2
Ruijie(config-if-GigabitEthernet 4/2)# switchport access vlan 10
Ruijie(config-if-GigabitEthernet 4/2)# exit
```

H-VPLS (QinQ Access and Dual-Homed)

If the U-PE and N-PE are connected through only one Spoke, the U-PE communication with the external network is discontinued if the Spoke connection fails. To solve this problem, you can connect the U-PE to N-PEs in dual-homed mode. In this mode, the U-PE is connected to different N-PEs, which belong to the same VPLS network, through two (or more) Spoke connections. In normal situations, only one of the Spoke connections is working. It is called the active connection and the others are backup ones. If the active connection fails, one backup connection is chosen to take over the task.

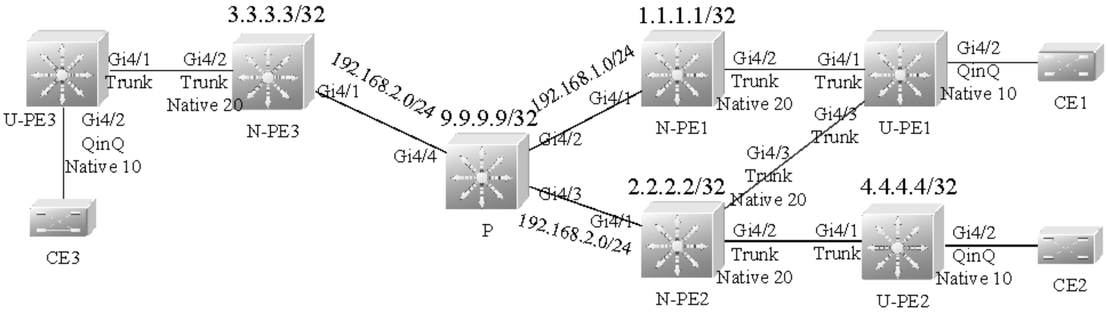
The following shows a schematic diagram of an H-VPLS network in dual-homed mode.

Figure 73



On an H-VPLS network in QinQ and dual-homed access mode, you can use the STP to switch over the active and standby connections. The STP, however, does not belong to users and is transmitted only over the carrier's network (U-PEs and N-PEs). To avoid the impact of STP protocol packets (BPDUs) on user packets, you must create a special VPLS instance on the N-PE to transparently transmit BPDU packets in dual-homed mode.

Figure 40



The preceding figure shows an H-VPLS topology in dual-homed mode. The CE is connected to U-PEs on the same H-VPLS network and the U-PEs are connected to N-PEs through the QinQ tunnel. The N-PEs belong to the MPLS network and the U-PEs do not run MPLS, but the N-PEs and U-PEs work together to provide VPLS services. The upstream interface on the U-PE works in trunk mode and allows the traffic of the VLAN to which the U-PE interface connected to the CE belongs to pass through. Enable QinQ on the interfaces that connect U-PEs and CEs. This QinQ native LVAN cannot be the same as the native VLAN on the U-PE1 interface connected to the N-PE or the native VLAN on the N-PE interface connected to the U-PE. This is because the QinQ native VLAN on the interfaces that connect U-PEs and CEs is used to identify VPLS instances provided for a user whereas the native VLAN on the N-PE trunk interface connected to the U-PE is used to identify the VPLS instance for transparently transmitting BPDU packets in dual-homed mode. Set up Hub PWs between N-PEs. Enable dual-homed mode on U-PE1 and you can find that U-PE1 is connected to both N-PE1 and N-PE2.

As shown in Figure 1-27, only one VPLS instance is provided for the user. You are required to create a special VPLS instance to transparently transmit BPDU packets in dual-homed mode. You only need to enable this VPLS instance on the N-PEs to which the U-PE is connected.

Configuring U-PE1:

# Configure a Spoke connection to the peer N-PE1.

```
Ruijie(config)# interface gigabitEthernet 4/1
Ruijie(config-if-GigabitEthernet 4/1)# switchport mode trunk
Ruijie(config-if-GigabitEthernet 4/1)# exit
```

# Configure a Spoke connection to the peer N-PE2.

```
Ruijie(config)# interface gigabitEthernet 4/3
Ruijie(config-if-GigabitEthernet 4/3)# switchport mode trunk
Ruijie(config-if-GigabitEthernet 4/3)# exit
```

# Configure the interface between PEs and CEs.

```
Ruijie(config)# interface gigabitEthernet 4/2
Ruijie(config-if-GigabitEthernet 4/2)# switchport access vlan 10
Ruijie(config-if-GigabitEthernet 4/2)# switchport mode dot1q-tunnel
Ruijie(config-if-GigabitEthernet 4/2)# exit
```

# Enable STP only on the U-PEs that adopt the dual-homed access mode.

```
Ruijie(config)# spanning-tree
```

Configuring N-PE1:

# Configure the loopback interface.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 1.1.1.1 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 1.1.1.1 0.0.0.0 area 0
Ruijie(config-router)# network 192.168.1.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# neighbor 2.2.2.2
Ruijie(config-mpls-router)# neighbor 3.3.3.3
Ruijie(config-mpls-router)# exit
```

# Enable LDP and MPLS on the public network interface.

```
Ruijie(config)# interface gigabitEthernet 4/1
Ruijie(config-if-GigabitEthernet 4/1)# no switchport
Ruijie(config-if-GigabitEthernet 4/1)# ip address 192.168.1.2 255.255.255.0
Ruijie(config-if-GigabitEthernet 4/1)# mpls ip
Ruijie(config-if-GigabitEthernet 4/1)# label-switching
Ruijie(config-if-GigabitEthernet 4/1)# exit
```

# Configure the N-PE interface connected to the U-PE.

```
Ruijie(config)# interface gigabitEthernet 4/2
Ruijie(config-if-GigabitEthernet 4/2)# switchport mode trunk
Ruijie(config-if-GigabitEthernet 4/2)# switchport trunk native vlan 20
Ruijie(config-if-GigabitEthernet 4/2)# exit
```

# Configure a VPLS instance for the user and enable Hub PWs to connect the N-PEs in the same VPLS instance.

```
Ruijie(config)# l2 vfi vfi_a vpnid 1
Ruijie(config-vpls)# neighbor 2.2.2.2 encapsulation mpls
Ruijie(config-vpls)# neighbor 3.3.3.3 encapsulation mpls
Ruijie(config-vpls)# exit
```

# Bind the QinQ VLAN interface to the VPLS instance.

```
Ruijie(config)# vlan 10
Ruijie(config-vlan)# exit
Ruijie(config)# interface vlan 10
Ruijie(config-if-Vlan 10)# xconnect vfi vfi_a
Ruijie(config-if-Vlan 10)# exit
```

# Configure the VPLS instance to transparently transmit BPDU packets and enable Hub PWs to connect other N-PEs in dual-homed mode.

```
Ruijie(config)# l2 vfi vfi_bpdu vpnid 100
Ruijie(config-vpls)# neighbor 2.2.2.2 encapsulation mpls
Ruijie(config-vpls)# exit
```

# Bind the native VLAN interface, which is used to transparently transmit BPDUs, to the VPLS instance.

```
Ruijie(config)# vlan 20
Ruijie(config-vlan)# exit
Ruijie(config)# interface vlan 20
Ruijie(config-if-Vlan 20)# xconnect vfi vfi_bpdu
Ruijie(config-if-Vlan 20)# l2 vfi tunnel-protocol stp
Ruijie(config-if-Vlan 20)# exit
```

Configuring P:

# Configure the loopback interface.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 9.9.9.9 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 9.9.9.9 0.0.0.0 area 0
Ruijie(config-router)# network 192.168.1.0 0.0.0.255  area 0
Ruijie(config-router)# network 192.168.2.0 0.0.0.255  area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# exit
```

# Enable LDP and MPLS on an interface.

```
Ruijie(config)# interface gigabitEthernet 4/2
Ruijie(config-if-GigabitEthernet 4/2)#  no switchport
Ruijie(config-if-GigabitEthernet 4/2)#  ip address 192.168.1.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 4/2)#  mpls ip
Ruijie(config-if-GigabitEthernet 4/2)#  label-switching
Ruijie(config-if-GigabitEthernet 4/2)#  exit
Ruijie(config)# interface gigabitEthernet 4/3
Ruijie(config-if-GigabitEthernet 4/3)#  no switchport
Ruijie(config-if-GigabitEthernet 4/3)#  ip address 192.168.2.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 4/3)#  mpls ip
Ruijie(config-if-GigabitEthernet 4/3)#  label-switching
Ruijie(config-if-GigabitEthernet 4/3)# exit
```

Configuring N-PE2:

# Configure the loopback interface.

```
Ruijie(config)#  interface loopback 0
Ruijie(config-if-Loopback 0)# # ip address 2.2.2.2 255.255.255.255
Ruijie(config-if-Loopback 0)#  exit
```

# Configure OSPF and establish public network routes.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 2.2.2.2 0.0.0.0  area 0
Ruijie(config-router)# network 192.168.2.0 0.0.0.255  area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# neighbor 1.1.1.1
Ruijie(config-mpls-router)# neighbor 3.3.3.3
Ruijie(config-mpls-router)# exit
```

# Enable LDP and MPLS on the public network interface.

```
Ruijie(config)# interface gigabitEthernet 4/1
Ruijie(config-if-GigabitEthernet 4/1)# no switchport
Ruijie(config-if-GigabitEthernet 4/1)# ip address 192.168.2.2 255.255.255.0
Ruijie(config-if-GigabitEthernet 4/1)# mpls ip
Ruijie(config-if-GigabitEthernet 4/1)# label-switching
Ruijie(config-if-GigabitEthernet 4/1)# exit
```

# Configure the N-PE interface connected to U-PEs (to two U-PEs at the same time).

```
Ruijie(config)# interface gigabitEthernet 4/2
Ruijie(config-if-GigabitEthernet 4/2)# switchport mode trunk
Ruijie(config-if-GigabitEthernet 4/2)# switchport trunk native vlan 20
Ruijie(config-if-GigabitEthernet 4/2)# exit
Ruijie(config)# interface gigabitEthernet 4/3
Ruijie(config-if-GigabitEthernet 4/3)# switchport mode trunk
Ruijie(config-if-GigabitEthernet 4/3)#  switchport trunk native vlan 20
Ruijie(config-if-GigabitEthernet 4/3)#  exit
```

# Configure a VPLS instance for the user and enable Hub PWs to connect the N-PEs in the same VPLS instance.

```
Ruijie(config)# l2 vfi vfi_a vpnid 1
Ruijie(config-vpls)#  neighbor 1.1.1.1 encapsulation mpls
Ruijie(config-vpls)# neighbor 3.3.3.3 encapsulation mpls
Ruijie(config-vpls)# exit
```

# Bind the QinQ VLAN interface to the VPLS instance.

```
Ruijie(config)# vlan 10
Ruijie(config-vlan)# exit
Ruijie(config)# interface vlan 10
Ruijie(config-if-Vlan 10)# xconnect vfi vfi_a
Ruijie(config-if-Vlan 10)# exit
```

# Enable the VPLS instance to transparently transmit BPDU packets and enable Hub PWs to connect other N-PEs in dual-homed mode.

```
Ruijie(config)# l2 vfi vfi_bpdu vpnid 100
Ruijie(config-vpls)# neighbor 1.1.1.1 encapsulation mpls
Ruijie(config-vpls)# exit
```

# Bind the native VLAN interface, which is used to transparently transmit BPDU packets, to the VPLS instance.

```
Ruijie(config)# vlan 20
Ruijie(config-vlan)# exit
Ruijie(config)# interface vlan 20
Ruijie(config-if-Vlan 20)# xconnect vfi vfi_bpdu
Ruijie(config-if-Vlan 20)# l2 vfi tunnel-protocol stp
Ruijie(config-if-Vlan 20)# exit
```

Configuring U-PE2:

# Refer to U-PE1. Since U-PE2 is connected to only one N-PE, STP is not required.

Configuring U-PE3:

# Refer to U-PE2.

Configuring N-PE3:

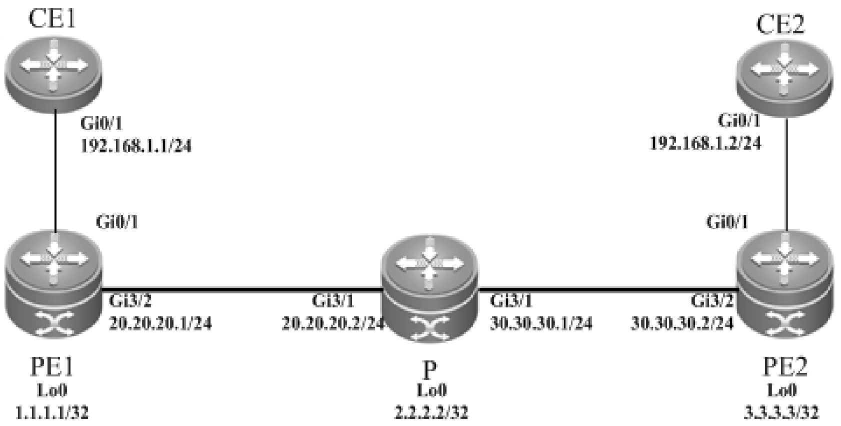
# Refer to N-PE1.

Typical Examples of Martini VPLS Configuration for Routers

Basic VPLS (Ethernet Interface Access)

As shown in the following figure, CE1 and CE2 access the same VPLS network through PE1 and PE2. PE1, P, and PE2 form a public MPLS network to provide VPLS services. Gi0/1 is bound on PE1 and PE2 to the VPLS instance.

Figure 41



The configuration procedure is as follows:

Configuring PE1:

# Configure the loopback interface address.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 1.1.1.1 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 1.1.1.1 0.0.0.0 area 0
Ruijie(config-router)# network 20.20.20.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# neighbor 3.3.3.3
Ruijie(config-mpls-router)# exit
```

# Enable LDP and MPLS on the public network interface.

```
Ruijie(config)# interface gigabitEthernet 3/2
Ruijie(config-if-Gigabitethernet 3/2)# ip ref
Ruijie(config-if-Gigabitethernet 3/2)# ip address 20.20.20.1 255.255.255.0
Ruijie(config-if-Gigabitethernet 3/2)# mpls ip
Ruijie(config-if-Gigabitethernet 3/2)# label-switching
Ruijie(config-if-Gigabitethernet 3/2)# exit
```

# Configure a VPLS instance and specify the peer PE.

```
Ruijie(config)# l2 vfi vfi_a vpnid 1
Ruijie(config-vpls)# neighbor 3.3.3.3 encapsulation mpls ethernet
Ruijie(config-vpls)# exit
```

# Configure the interface that connects PEs and CEs to bind the VPLS instance.

```
Ruijie(config)# interface gigabitethernet 0/1
Ruijie(config-if-Gigabitethernet 0/1)# ip ref
Ruijie(config-if-Gigabitethernet 0/1)# xconnect vfi vfi_a
Ruijie(config-if-Gigabitethernet 0/1)# exit
```

Configuring P:

# Configure the loopback interface address.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 2.2.2.2 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 2.2.2.2 0.0.0.0 area 0
Ruijie(config-router)# network 20.20.20.0 0.0.0.255 area 0
Ruijie(config-router)# network 30.30.30.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# exit
```

# Enable LDP and MPLS on an interface.

```
Ruijie(config)# interface gigabitEthernet 3/1
Ruijie(config-if-Gigabitethernet 3/1)# ip ref
Ruijie(config-if-Gigabitethernet 3/1)# ip address 20.20.20.2 255.255.255.0
Ruijie(config-if-Gigabitethernet 3/1)# mpls ip
Ruijie(config-if-Gigabitethernet 3/1)# label-switching
Ruijie(config-if-Gigabitethernet 3/1)# exit
Ruijie(config)# interface gigabitEthernet 3/2
Ruijie(config-if-Gigabitethernet 3/2)# ip ref
Ruijie(config-if-Gigabitethernet 3/2)# ip address 30.30.30.1 255.255.255.0
Ruijie(config-if-Gigabitethernet 3/2)# mpls ip
Ruijie(config-if-Gigabitethernet 3/2)# label-switching
Ruijie(config-if-Gigabitethernet 3/2)# exit
```

Configuring PE2:

# Configure the loopback interface address.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 3.3.3.3 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 3.3.3.3 0.0.0.0 area 0
Ruijie(config-router)# network 30.30.30.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# neighbor 1.1.1.1
Ruijie(config-mpls-router)# exit
```

# Enable LDP and MPLS on the public network interface.

```
Ruijie(config)# interface gigabitEthernet 3/1
Ruijie(config-if-Gigabitethernet 3/1)# ip ref
Ruijie(config-if-Gigabitethernet 3/1)# ip address 30.30.30.2 255.255.255.0
Ruijie(config-if-Gigabitethernet 3/1)# mpls ip
Ruijie(config-if-Gigabitethernet 3/1)# label-switching
Ruijie(config-if-Gigabitethernet 3/1)# exit
```

# Configure a VPLS instance and specify the peer PE.

```
Ruijie(config)# l2 vfi vfi_a vpnid 1
Ruijie(config-vpls)# neighbor 1.1.1.1 encapsulation mpls ethernet
Ruijie(config-vpls)# exit
```

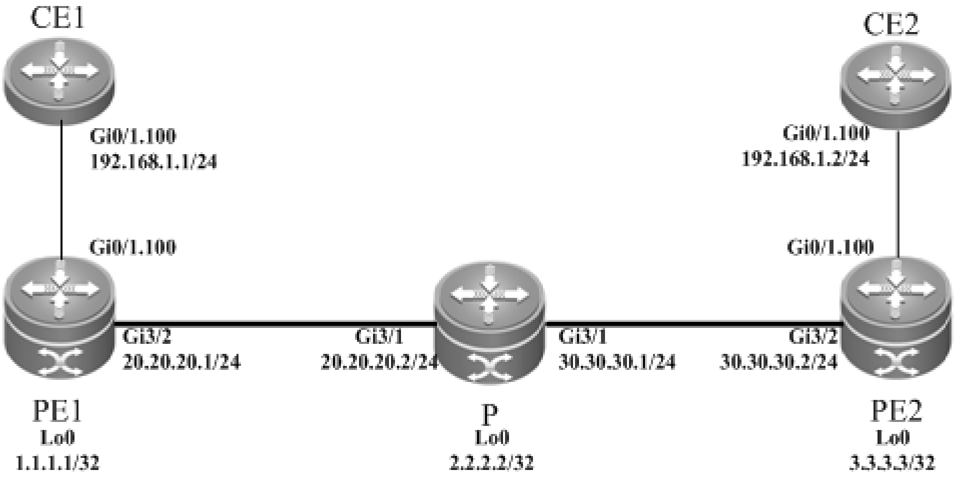
# Configure the interface that connects PEs and CEs to bind the VPLS instance.

```
Ruijie(config)# interface gigabitethernet 0/1
Ruijie(config-if-Gigabitethernet 0/1)# ip ref
Ruijie(config-if-Gigabitethernet 0/1)# xconnect vfi vfi_a
Ruijie(config-if-Gigabitethernet 0/1)# exit
```

Basic VPLS (Ethernet Sub-interface Access)

As shown in the following figure, CE1 and CE2 access the same VPLS network through PE1 and PE2. PE1, P, and PE2 form a public MPLS network to provide VPLS services. VLAN 10 is bound on PE1 and PE2 to the VPLS instance.

Figure 74



The configuration procedure is as follows:

Configuring PE1:

# Configure the loopback interface address.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 1.1.1.1 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 1.1.1.1 0.0.0.0 area 0
Ruijie(config-router)# network 20.20.20.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# neighbor 3.3.3.3
Ruijie(config-mpls-router)# exit
```

# Enable LDP and MPLS on the public network interface.

```
Ruijie(config)# interface gigabitEthernet 3/2
Ruijie(config-if-Gigabitethernet 3/2)# ip ref
Ruijie(config-if-Gigabitethernet 3/2)# ip address 20.20.20.1 255.255.255.0
Ruijie(config-if-Gigabitethernet 3/2)# mpls ip
Ruijie(config-if-Gigabitethernet 3/2)# label-switching
Ruijie(config-if-Gigabitethernet 3/2)# exit
```

# Configure a VPLS instance and specify the peer PE.

```
Ruijie(config)# l2 vfi vfi_a vpnid 1
Ruijie(config-vpls)# neighbor 3.3.3.3 encapsulation mpls ethernetvlan
Ruijie(config-vpls)# exit
```

# Configure the interface that connects PEs and CEs to bind the VPLS instance.

```
Ruijie(config)# interface gigabitethernet 0/1
Ruijie(config-if-Gigabitethernet 0/1)# ip ref
Ruijie(config-if-Gigabitethernet 0/1)# exit
Ruijie(config)# interface gigabitethernet 0/1.100
Ruijie(config-if-Gigabitethernet 0/1.100)# encapsulation dot1Q 100
Ruijie(config-if-Gigabitethernet 0/1.100)# xconnect vfi vfi_a
Ruijie(config-if-Gigabitethernet 0/1.100)# exit
```

Configuring P:

# Configure the loopback interface address.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 2.2.2.2 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 2.2.2.2 0.0.0.0 area 0
Ruijie(config-router)# network 20.20.20.0 0.0.0.255 area 0
Ruijie(config-router)# network 30.30.30.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# exit
```

# Enable LDP and MPLS on an interface.

```
Ruijie(config)# interface gigabitEthernet 3/1
Ruijie(config-if-Gigabitethernet 3/1)# ip ref
Ruijie(config-if-Gigabitethernet 3/1)# ip address 20.20.20.2 255.255.255.0
Ruijie(config-if-Gigabitethernet 3/1)# mpls ip
Ruijie(config-if-Gigabitethernet 3/1)# label-switching
Ruijie(config-if-Gigabitethernet 3/1)# exit
Ruijie(config)# interface gigabitEthernet 3/2
Ruijie(config-if-Gigabitethernet 3/2)# ip ref
Ruijie(config-if-Gigabitethernet 3/2)# ip address 30.30.30.1 255.255.255.0
Ruijie(config-if-Gigabitethernet 3/2)# mpls ip
Ruijie(config-if-Gigabitethernet 3/2)# label-switching
Ruijie(config-if-Gigabitethernet 3/2)# exit
```

Configuring PE2:

# Configure the loopback interface address.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 3.3.3.3 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 3.3.3.3 0.0.0.0 area 0
Ruijie(config-router)# network 30.30.30.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# neighbor 1.1.1.1
Ruijie(config-mpls-router)# exit
```

# Enable LDP and MPLS on the public network interface.

```
Ruijie(config)# interface gigabitEthernet 3/1
Ruijie(config-if-Gigabitethernet 3/1)# ip ref
Ruijie(config-if-Gigabitethernet 3/1)# ip address 30.30.30.2 255.255.255.0
Ruijie(config-if-Gigabitethernet 3/1)# mpls ip
Ruijie(config-if-Gigabitethernet 3/1)# label-switching
Ruijie(config-if-Gigabitethernet 3/1)# exit
```

# Configure a VPLS instance and specify the peer PE.

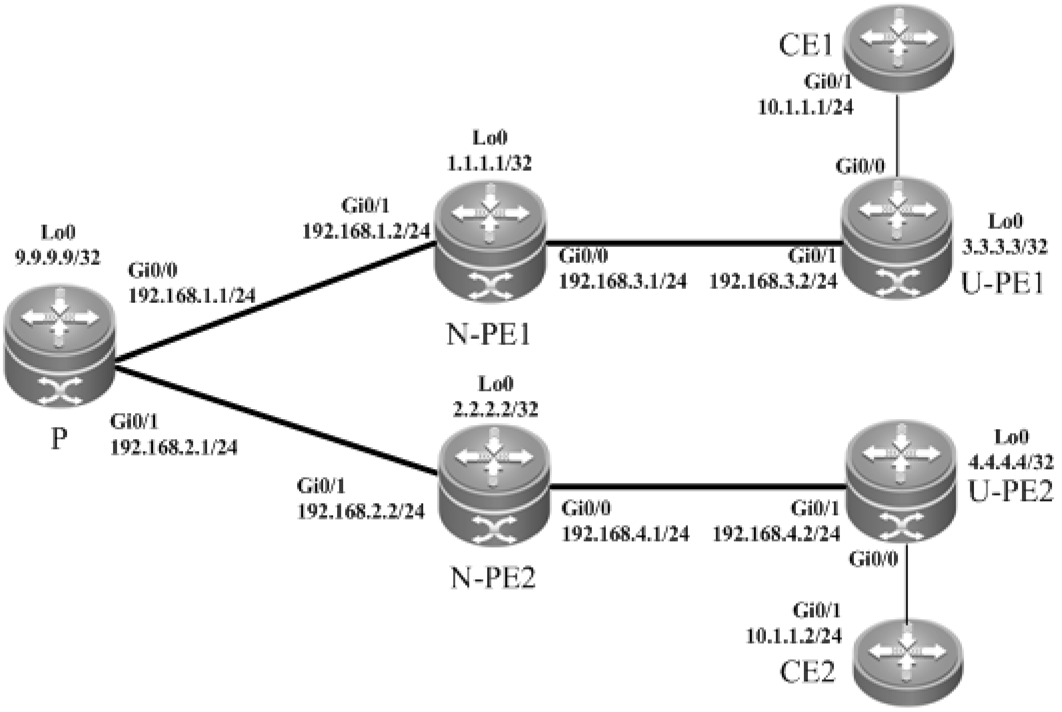
```
Ruijie(config)# l2 vfi vfi_a vpnid 1
Ruijie(config-vpls)# neighbor 1.1.1.1 encapsulation mpls ethernetvlan
Ruijie(config-vpls)# exit
```

# Configure the interface that connects PEs and CEs to bind the VPLS instance.

```
Ruijie(config)# interface gigabitethernet 0/1
Ruijie(config-if-Gigabitethernet 0/1)# ip ref
Ruijie(config-if-Gigabitethernet 0/1)# exit
Ruijie(config)# interface gigabitethernet 0/1.100
Ruijie(config-if-Gigabitethernet 0/1.100)# encapsulation dot1Q 100
Ruijie(config-if-Gigabitethernet 0/1.100)# xconnect vfi vfi_a
Ruijie(config-if-Gigabitethernet 0/1.100)# exit
```

H-VPLS (PW Access)

Figure 75



As shown in the preceding figure, CE1 and CE2 access the same H-VPLS network through U-PE1 and U-PE2, which are connected to N-PE1 and N-PE2 respectively. The N-PEs and U-PEs are connected through PWs. All U-PEs and N-PEs belong to the MPLS network and work together to provide VPLS services.

When setting up PWs between N-PEs and U-PEs, you must specify the PW of the N-PEs with the U-PEs as a Spoke PW. On the U-PE end, the PW type is not restricted. If U-PEs support VPLS, the PW between U-PEs and N-PEs can be either a Hub PW or a Spoke PW. If the U-PEs do not support VPLS, the VPWS PW can also be configured.

In this example, U-PE1 and N-PE1 are connected through a Hub PW whereas U-PE2 and N-PE2 are connected through a VPWS PW.

Configuring U-PE1:

# Configure the loopback interface address.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 3.3.3.3 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 3.3.3.3 0.0.0.0 area 0
Ruijie(config-router)# network 192.168.3.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# neighbor 1.1.1.1
Ruijie(config-mpls-router)# exit
```

# Enable LDP and MPLS on the public network interface.

```
Ruijie(config)# interface gigabitEthernet 0/1
Ruijie(config-if-GigabitEthernet 0/1)# ip ref
Ruijie(config-if-GigabitEthernet 0/1)# ip address 192.168.3.2 255.255.255.0
Ruijie(config-if-GigabitEthernet 0/1)# mpls ip
Ruijie(config-if-GigabitEthernet 0/1)# label-switching
Ruijie(config-if-GigabitEthernet 0/1)# exit
```

# Configure a VPLS instance and enable PWs to connect to the peer N-PE

```
Ruijie(config)# l2 vfi vfi_a vpnid 1
Ruijie(config-vpls)# neighbor 1.1.1.1 encapsulation mpls
Ruijie(config-vpls)# exit
```

# Configure the interface that connects PEs and CEs to bind the VPLS instance.

```
Ruijie(config)# interface gigabitEthernet 0/0
Ruijie(config-if-GigabitEthernet 0/0)# ip ref
Ruijie(config-if-GigabitEthernet 0/0)# exit
```

Configuring N-PE1:

# Configure the loopback interface address.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 1.1.1.1 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 1.1.1.1 0.0.0.0 area 0
Ruijie(config-router)# network 192.168.1.0 0.0.0.255 area 0
Ruijie(config-router)# network 192.168.3.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# neighbor 2.2.2.2
Ruijie(config-mpls-router)# neighbor 3.3.3.3
Ruijie(config-mpls-router)# exit
```

# Enable LDP and MPLS on the public network interface.

```
Ruijie(config)# interface gigabitEthernet 0/1
Ruijie(config-if-GigabitEthernet 0/1)# ip ref
Ruijie(config-if-GigabitEthernet 0/1)# ip address 192.168.1.2 255.255.255.0
Ruijie(config-if-GigabitEthernet 0/1)# mpls ip
Ruijie(config-if-GigabitEthernet 0/1)# label-switching
Ruijie(config-if-GigabitEthernet 0/1)# exit
Ruijie(config)# interface gigabitEthernet 0/0
Ruijie(config-if-GigabitEthernet 0/0)# ip ref
Ruijie(config-if-GigabitEthernet 0/0)# ip address 192.168.3.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 0/0)# mpls ip
Ruijie(config-if-GigabitEthernet 0/0)# label-switching
Ruijie(config-if-GigabitEthernet 0/0)# exit
```

# Configure a VPLS instance, enable a Hub PW to connect another N-PE, and enable a Spoke PW to connect the peer U-PE.

```
Ruijie(config)# l2 vfi vfi_a vpnid 1
Ruijie(config-vpls)# neighbor 2.2.2.2 encapsulation mpls
Ruijie(config-vpls)# neighbor 3.3.3.3 encapsulation mpls spoke-vc
Ruijie(config-vpls)# exit
```

Configuring P:

# Configure the loopback interface address.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 9.9.9.9 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 9.9.9.9 0.0.0.0 area 0
Ruijie(config-router)# network 192.168.1.0 0.0.0.255 area 0
Ruijie(config-router)# network 192.168.2.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# exit
```

# Enable LDP and MPLS on an interface.

```
Ruijie(config)# interface gigabitEthernet 0/0
Ruijie(config-if-GigabitEthernet 0/0)# ip ref
Ruijie(config-if-GigabitEthernet 0/0)# ip address 192.168.1.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 0/0)# mpls ip
Ruijie(config-if-GigabitEthernet 0/0)# label-switching
Ruijie(config-if-GigabitEthernet 0/0)# exit
Ruijie(config)# interface gigabitEthernet 0/1
Ruijie(config-if-GigabitEthernet 0/1)# ip ref
Ruijie(config-if-GigabitEthernet 0/1)# ip address 192.168.2.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 0/1)# mpls ip
Ruijie(config-if-GigabitEthernet 0/1)# label-switching
Ruijie(config-if-GigabitEthernet 0/1)# exit
```

Configuring N-PE2:

# Configure the loopback interface address.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 2.2.2.2 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 2.2.2.2 0.0.0.0 area 0
Ruijie(config-router)# network 192.168.2.0 0.0.0.255 area 0
Ruijie(config-router)# network 192.168.4.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# neighbor 1.1.1.1
Ruijie(config-mpls-router)# neighbor 4.4.4.4
Ruijie(config-mpls-router)# exit
```

# Enable LDP and MPLS on the public network interface.

```
Ruijie(config)# interface gigabitEthernet 0/1
Ruijie(config-if-GigabitEthernet 0/1)# ip ref
Ruijie(config-if-GigabitEthernet 0/1)# ip address 192.168.2.2 255.255.255.0
Ruijie(config-if-GigabitEthernet 0/1)# mpls ip
Ruijie(config-if-GigabitEthernet 0/1)# label-switching
Ruijie(config-if-GigabitEthernet 0/1)# exit
Ruijie(config)# interface gigabitEthernet 0/0
Ruijie(config-if-GigabitEthernet 0/0)# ip ref
Ruijie(config-if-GigabitEthernet 0/0)# ip address 192.168.4.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 0/0)# mpls ip
Ruijie(config-if-GigabitEthernet 0/0)# label-switching
Ruijie(config-if-GigabitEthernet 0/0)# exit
```

# Configure a VPLS instance, enable a Hub PW to connect another N-PE, and enable a Spoke PW to connect the peer U-PE.

```
Ruijie(config)# l2 vfi vfi_a vpnid 1
Ruijie(config-vpls)# neighbor 1.1.1.1 encapsulation mpls
Ruijie(config-vpls)# neighbor 4.4.4.4 encapsulation mpls  spoke-vc
Ruijie(config-vpls)# exit
```

Configuring U-PE2:

# Configure the loopback interface address.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 4.4.4.4 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 4.4.4.4 0.0.0.0 area 0
Ruijie(config-router)# network 192.168.4.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# neighbor 2.2.2.2
Ruijie(config-mpls-router)# exit
```

# Enable LDP and MPLS on the public network interface.

```
Ruijie(config)# interface gigabitEthernet 0/1
Ruijie(config-if-GigabitEthernet 0/1)# ip ref
Ruijie(config-if-GigabitEthernet 0/1)# ip address 192.168.4.2 255.255.255.0
Ruijie(config-if-GigabitEthernet 0/1)#mpls ip
Ruijie(config-if-GigabitEthernet 0/1)# label-switching
Ruijie(config-if-GigabitEthernet 0/1)#  exit
```

# Configure the interface that connects PEs and CEs to enable VPWS.

```
Ruijie(config)# interface gigabitEthernet 0/0
Ruijie(config-if-GigabitEthernet 0/0)# ip ref
Ruijie(config-if-GigabitEthernet 0/0)# xconnect 2.2.2.2 1 encapsulation mpls ethernet
Ruijie(config-if-GigabitEthernet 0/0)# exit
```

Typical Examples of Kompella VPLS Configuration for Switches

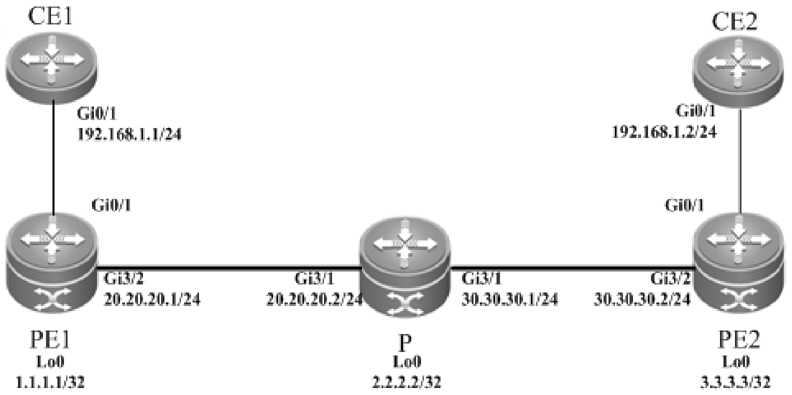
Basic VPLS

Networking Requirements

- CE1 and CE2 access the same VPLS network through PE1 and PE2.
- PE1, P and PE2 form the public MPLS network.
- VLAN 10 is bound on PE1 and PE2 to the VPLS instance.

Networking Topology

Figure 76



Configuration Tips

Before configuring Kompella VPLS, complete the following tasks:

- Run IGP in the carrier's network to realize connection between PE1 and PE2.
- Obtain Kompella VPLS configuration information including VPLS instance descriptive information, RT value, VE ID, maximum planned site number, VE ID deviation, and interface information from the network administrator.

Configuration Steps

Configuring PE1:

# Configure the loopback interface address.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 1.1.1.1 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 1.1.1.1 0.0.0.0 area 0
Ruijie(config-router)# network 20.20.20.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# neighbor 3.3.3.3
Ruijie(config-mpls-router)# exit
```

# Enable LDP and MPLS on the public network interface.

```
Ruijie(config)# interface gigabitEthernet 3/2
Ruijie(config-if-Gigabitethernet 3/2)# no switchport
Ruijie(config-if-Gigabitethernet 3/2)# ip address 20.20.20.1 255.255.255.0
Ruijie(config-if-Gigabitethernet 3/2)# mpls ip
Ruijie(config-if-Gigabitethernet 3/2)# label-switching
Ruijie(config-if-Gigabitethernet 3/2)# exit
```

# Configure a VLAN interface.

```
Ruijie(config)# vlan 10
Ruijie(config-vlan)# exit
```

# Configure the L2VPN address family.

```
Ruijie(config)# router bgp 100
Ruijie(config-router)# neighbor 3.3.3.3 remote-as 100
Ruijie(config-router)# neighbor 3.3.3.3 update-source loopback 0
Ruiije(config-router)# address-family l2vpn vpls
Ruijie(config-router-af)# neighbor 3.3.3.3 activate
Ruijie(config-router-af)# neighbor 3.3.3.3 send-community extended
Ruijie(config-router-af)# exit
```

# Configure the interface that connects PEs and CEs.

```
Ruijie(config)# interface gigabitEthernet 3/1
Ruijie(config-if-GigabitEthernet 3/1)# switchport access vlan 10
Ruijie(config-if-GigabitEthernet 3/1)# exit
```

# Configure a VPLS instance.

```
Ruijie# configure terminal
Ruijie(config)# l2 vfi vpls-1 vpnid 1 autodiscovery
Ruijie(config-vfi)# rd 1:1
Ruijie(config-vfi)# signal bgp
Ruijie(config-vfi)# encapsulation mpls ethernet
Ruijie(config-vfi)# route-target both 10000:1
Ruijie(config-vfi)# site-id 1
Ruijie(config-vfi-site)# xconnect interface vlan 10
Ruijie(config-vfi-site)#exit-site-mode
Ruijie(config-vfi)# eixt
```

Configuring P:

# Configure the loopback interface address.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 2.2.2.2 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 2.2.2.2 0.0.0.0 area 0
Ruijie(config-router)# network 20.20.20.0 0.0.0.255 area 0
Ruijie(config-router)# network 30.30.30.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# exit
```

# Enable LDP and MPLS on an interface.

```
Ruijie(config)# interface gigabitEthernet 3/1
Ruijie(config-if-Gigabitethernet 3/1)# no switchport
Ruijie(config-if-Gigabitethernet 3/1)# ip address 20.20.20.2 255.255.255.0
Ruijie(config-if-Gigabitethernet 3/1)# mpls ip
Ruijie(config-if-Gigabitethernet 3/1)# label-switching
Ruijie(config-if-Gigabitethernet 3/1)# exit
Ruijie(config)# interface gigabitEthernet 3/2
Ruijie(config-if-Gigabitethernet 3/2)# no switchport
Ruijie(config-if-Gigabitethernet 3/2)# ip address 30.30.30.1 255.255.255.0
Ruijie(config-if-Gigabitethernet 3/2)# mpls ip
Ruijie(config-if-Gigabitethernet 3/2)# label-switching
Ruijie(config-if-Gigabitethernet 3/2)# exit
```

Configuring PE2:

# Configure the loopback interface address.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 3.3.3.3 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 3.3.3.3 0.0.0.0 area 0
Ruijie(config-router)# network 30.30.30.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# neighbor 1.1.1.1
Ruijie(config-mpls-router)# exit
```

# Enable LDP and MPLS on the public network interface.

```
Ruijie(config)# interface gigabitEthernet 3/1
Ruijie(config-if-Gigabitethernet 3/1)# no switchport
Ruijie(config-if-Gigabitethernet 3/1)# ip address 30.30.30.2 255.255.255.0
Ruijie(config-if-Gigabitethernet 3/1)# mpls ip
Ruijie(config-if-Gigabitethernet 3/1)# label-switching
Ruijie(config-if-Gigabitethernet 3/1)# exit
```

# Configure the L2VPN address family.

```
Ruijie(config)# router bgp 100
Ruijie(config-router)# neighbor 1.1.1.1 remote-as 100
Ruijie(config-router)# neighbor 1.1.1.1 update-source loopback 0
Ruiije(config-router)# address-family l2vpn vpls
Ruijie(config-router-af)# neighbor 1.1.1.1 activate
Ruijie(config-router-af)# neighbor 1.1.1.1 send-community extended
Ruijie(config-router-af)# exit
```

# Configure a VLAN interface.

```
Ruijie(config)# vlan 10
Ruijie(config-vlan)# exit
```

# Configure the interface that connects PEs and CEs.

```
Ruijie(config)# interface gigabitEthernet 3/2
Ruijie(config-if-Gigabitethernet 3/2)#  switchport access vlan 10
Ruijie(config-if-Gigabitethernet 3/2)# exit
```

# Configure a VPLS instance.

```
Ruijie# configure terminal
Ruijie(config)# l2 vfi vpls-1 vpnid 1 autodiscovery
Ruijie(config-vfi)# rd 1:1
Ruijie(config-vfi)# signal bgp
Ruijie(config-vfi)# encapsulation mpls ethernet
Ruijie(config-vfi)# route-target both 10000:1
Ruijie(config-vfi)# site-id 2
Ruijie(config-vfi-site)# xconnect interface vlan 10
Ruijie(config-vfi-site)# exit-site-mode
Ruijie(config-vfi)# exit
```

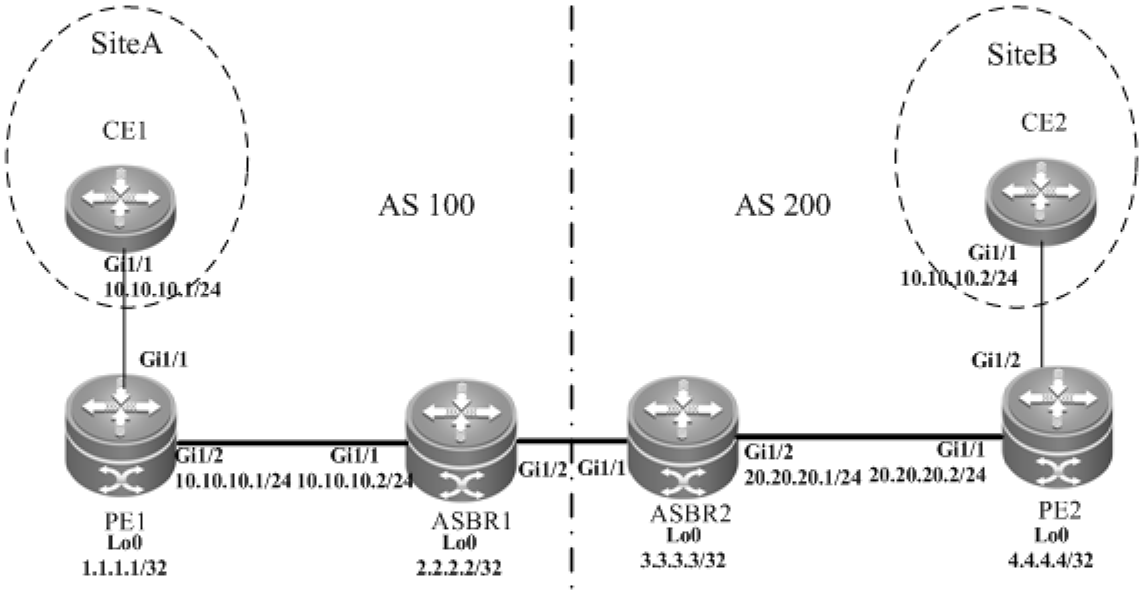
Inter-AS Configuration Examples – Option A Solution

Networking Requirements

- LAN segments of customer S in site A and site B are connected with each other through the carrier's PE1 in AS100 and PE2 in AS 200, forming a virtual and simulative LAN service, or VPLS service.
- PE1 and PE2 are in different ASs. ASBR1 and ASBR2 are considered CE devices by each other, which means that the interface between ASBRs connects the AC to the VPLS instance;

Networking Topology

Figure 77 Kompella VPLS inter-AS networking topology



The preceding figure shows the structure of the Kompella VPLS inter-AS networking topology in Option A. The intermediate interface is considered by ASBRs as AC connection.

Configuration Tips

Before configuring Kompella VPLS, complete the following tasks:

- Run IGP in the carrier's network to realize connection between PE and ASBR devices.
- Establish the MP-IBGP peer relationship between PEs and intra-AS ASBRs.
- Obtain Kompella VPLS configuration information including VPLS instance descriptive information, RT value, VE ID, maximum planned site number, VE ID deviation, and interface information from the network administrator.

Configuration Steps

- Configuring CE1

See "Configuring CE1" in basic configuration examples.

- Configuring PE1:

# Configure the loopback interface address.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 1.1.1.1 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes so that PEs can ping with ASBRs in the same AS.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 1.1.1.1 0.0.0.0 area 0
Ruijie(config-router)# network 10.10.10.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# exit
```

# Configure the public network tunnel between PEs.

```
Ruijie(config)# interface gigabitEthernet 1/2
Ruijie(config-if-Gigabitethernet 1/2)# ip address 10.10.10.1 255.255.255.0
Ruijie(config-if-Gigabitethernet 1/2)# mpls ip
Ruijie(config-if-Gigabitethernet 1/2)# label-switching
Ruijie(config-if-Gigabitethernet 1/2)# exit
```

# Configure the L2VPN address family.

```
Ruijie(config)# router bgp 100
Ruijie(config-router)# neighbor 2.2.2.2 remote-as 100
Ruijie(config-router)# neighbor 2.2.2.2 update-source loopback 0
Ruiije(config-router)# address-family l2vpn vpls
Ruijie(config-router-af)# neighbor 2.2.2.2 activate
Ruijie(config-router-af)# neighbor 2.2.2.2 send-community extended
Ruijie(config-router-af)# exit
```

# Configure a VLAN interface.

```
Ruijie(config)# vlan 10
Ruijie(config-vlan)# exit
```

# Configure the interface between CEs and PEs.

```
Ruijie(config)# interface gigabitEthernet 1/1
Ruijie(config-if-GigabitEthernet 1/1)# switchport access vlan 10
Ruijie(config-if-GigabitEthernet 1/1)# exit
```

# Configure a VPLS instance.

```
Ruijie# configure terminal
Ruijie(config)# l2 vfi vpls-1 vpnid 1 autodiscovery
Ruijie(config-vfi)# rd 100:1
Ruijie(config-vfi)# signal bgp
Ruijie(config-vfi)# encapsulation mpls ethernet
Ruijie(config-vfi)# route-target both 10000:1
Ruijie(config-vfi)# site-id 1
Ruijie(config-vfi-site)# xconnect interface vlan 10
Ruijie(config-vfi-site)#exit-site-mode
```

- Configuring ASBR1:

# Configure the loopback interface address.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 2.2.2.2 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF protocol and establish public network routes so that PEs can ping with ASBRs in the same AS.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 2.2.2.2 0.0.0.0 area 0
Ruijie(config-router)# network 10.10.10.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# exit
```

#Configure the public network tunnel between PEs.

```
Ruijie(config)# interface gigabitEthernet 1/1
Ruijie(config-if-Gigabitethernet 1/1)# ip address 10.10.10.2 255.255.255.0
Ruijie(config-if-Gigabitethernet 1/1)# mpls ip
Ruijie(config-if-Gigabitethernet 1/1)# label-switching
Ruijie(config-if-Gigabitethernet 1/1)# exit
```

# Configure the L2VPN address family.

```
Ruijie(config)# router bgp 100
Ruijie(config-router)# neighbor 1.1.1.1 remote-as 100
Ruijie(config-router)# neighbor 1.1.1.1 update-source loopback 0
Ruiije(config-router)# address-family l2vpn vpls
Ruijie(config-router-af)# neighbor 1.1.1.1 activate
Ruijie(config-router-af)# neighbor 1.1.1.1 send-community extended
Ruijie(config-router-af)# exit
```

# Configure a VLAN interface.

```
Ruijie(config)# vlan 10
Ruijie(config-vlan)# exit
```

# Configure the interface between ASBR1 and ASBR2.

```
Ruijie(config)# interface gigabitEthernet 1/2
Ruijie(config-if-GigabitEthernet 1/2)# switchport access vlan 10
Ruijie(config-if-GigabitEthernet 1/2)# exit
```

# Configure a VPLS instance.

```
Ruijie# configure terminal
Ruijie(config)# l2 vfi vpls-1 vpnid 1 autodiscovery
Ruijie(config-vfi)# rd 100:1
Ruijie(config-vfi)# signal bgp
Ruijie(config-vfi)# encapsulation mpls ethernet
Ruijie(config-vfi)# route-target both 10000:1
Ruijie(config-vfi)# mtu 1500
Ruijie(config-vfi)# site-id 2
Ruijie(config-vfi-site)# xconnect interface vlan 10
Ruijie(config-vfi-site)#exit-site-mode
Ruijie(config-vfi)#exit
```

- Configuring ASBR2:

# Configure the loopback interface address.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 3.3.3.3 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes so that PEs can ping with ASBRs in the same AS.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 3.3.3.3 0.0.0.0 area 0
Ruijie(config-router)# network 20.20.20.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# exit
```

# Configure the public network tunnel between PEs.

```
Ruijie(config)# interface gigabitEthernet 1/2
Ruijie(config-if-Gigabitethernet 1/2)# ip address 20.20.20.2 255.255.255.0
Ruijie(config-if-Gigabitethernet 1/2)# mpls ip
Ruijie(config-if-Gigabitethernet 1/2)# label-switching
Ruijie(config-if-Gigabitethernet 1/2)# exit
```

# Configure the L2VPN address family.

```
Ruijie(config)# router bgp 200
Ruijie(config-router)# neighbor 4.4.4.4 remote-as 200
Ruijie(config-router)# neighbor 4.4.4.4 update-source loopback 0
Ruiije(config-router)# address-family l2vpn vpls
Ruijie(config-router-af)# neighbor 4.4.4.4 activate
Ruijie(config-router-af)# neighbor 4.4.4.4 send-community extended
Ruijie(config-router-af)# exit
```

# Configure a VLAN interface.

```
Ruijie(config)# vlan 10
Ruijie(config-vlan)# exit
```

# Configure the interface between ASBR1 and ASBR2.

```
Ruijie(config)# interface gigabitEthernet 1/1
Ruijie(config-if-GigabitEthernet 1/1)#  switchport access vlan 10
Ruijie(config-if-GigabitEthernet 1/1)# exit
```

# Configure a VPLS instance.

```
Ruijie# configure terminal
Ruijie(config)# l2 vfi vpls-1 vpnid 1 autodiscovery
Ruijie(config-vfi)# rd 200:1
Ruijie(config-vfi)# signal bgp
Ruijie(config-vfi)# encapsulation mpls ethernet
Ruijie(config-vfi)# route-target both 10000:1
Ruijie(config-vfi)# site-id 3
Ruijie(config-vfi-site)# xconnect interface vlan 10
Ruijie(config-vfi-site)#exit-site-mode
Ruijie(config-vfi)#exit
```

- Configuring PE2:

# Configure the loopback interface address.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 4.4.4.4 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes so that PEs can ping with ASBRs in the same AS.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 4.4.4.4 0.0.0.0 area 0
Ruijie(config-router)# network 20.20.20.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# exit
```

# Configure the public network tunnel between PEs.

```
Ruijie(config)# interface gigabitEthernet 1/1
Ruijie(config-if-Gigabitethernet 1/1)# ip address 20.20.20.1 255.255.255.0
Ruijie(config-if-Gigabitethernet 1/1)# mpls ip
Ruijie(config-if-Gigabitethernet 1/1)# label-switching
Ruijie(config-if-Gigabitethernet 1/1)# exit
```

# Configure the L2VPN address family.

```
Ruijie(config)# router bgp 200
Ruijie(config-router)# neighbor 3.3.3.3 remote-as 200
Ruijie(config-router)# neighbor 3.3.3.3 update-source loopback 0
Ruiije(config-router)# address-family l2vpn vpls
Ruijie(config-router-af)# neighbor 3.3.3.3 activate
Ruijie(config-router-af)# neighbor 3.3.3.3 send-community extended
Ruijie(config-router-af)# exit
```

# Configure a VLAN interface.

```
Ruijie(config)# vlan 10
Ruijie(config-vlan)# exit
```

# Configure the interface that connects PE2 and a CE.

```
Ruijie(config)# interface gigabitEthernet 1/2
Ruijie(config-if-GigabitEthernet 1/2)# switchport access vlan 10
Ruijie(config-if-GigabitEthernet 1/2)# exit
```

# Configure a VPLS instance.

```
Ruijie# configure terminal
Ruijie(config)# l2 vfi vpls-1 vpnid 1 autodiscovery
Ruijie(config-vfi)# rd 200:1
Ruijie(config-vfi)# signal bgp
Ruijie(config-vfi)# encapsulation mpls ethernet
Ruijie(config-vfi)# route-target both 10000:1
Ruijie(config-vfi)# site-id 4
Ruijie(config-vfi-site)# xconnect interface vlan 10
Ruijie(config-vfi-site)#exit-site-mode
Ruijie(config-vfi)#exit
```

- Configuring CE2:

See "Configuring CE2" in basic configuration examples.

Verification

After the configuration, CE1 can ping with CE2.

After completing the configuration of Kompella VPLS, use the following commands to check the operation of VPLS.

Command	Function
Ruijie# **show bgp l2vpn vpls all**	Displays all the VPLS information.
Ruijie# **show mpls l2transport vc** [*vc_id* [*ip-address*]] | [**interface** *interface_name*] [**detail**]	Displays information about the PW (including VPWS PW and VPLS PW).
Ruijie# **show bgp l2vpn** { **vpls | vpws** } **all connections** [**neighbor** *address*] | [**interface** *interface_name*] | [**site-id** *id*] [**detail**]	Displays VPLS PW information.
Ruijie# **show mpls vfi** [*name*]	Displays all the configured or specified VFI information.

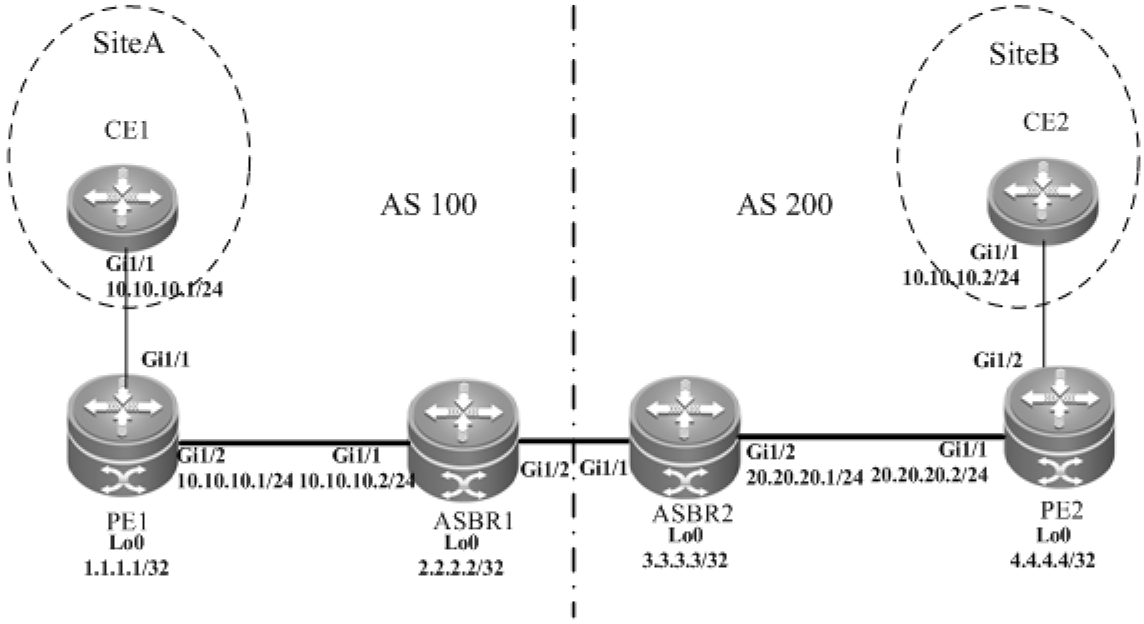
Inter-AS Configuration Examples – Option C Solution

Networking Requirements

- LAN segments of customer S in site A and site B are connected with each other through the carrier's PE1 in AS 1 and PE2 in AS 2, forming a virtual and simulative LAN service or VPLS service.
- PE1 and PE2 are in different ASs and can automatically detect PE devices involved in the VPLS instance.
- ASBR is not responsible for maintaining VPLS label block messages.
- VPLS label block messages are directly exchanged between PEs.

Networking Topology

Figure 78 Kompella VPLS Option C inter-AS networking topology



The preceding figure shows the structure of Kompella VPLS Option C inter-AS networking topology. Customer S's LAN segments in site A and site B are connected to each other through PE1 in AS 100 and PE2 in AS200 as one LAN.

Configuration Tips

Before configuring Kompella VPLS, complete the following tasks:

- Run IGP in the carrier's network to realize connection between VPLS-PE and ASBR devices in the same AS.
- Establish a public network tunnel between PE and ASBR devices in the same AS and enable MPLS on the ASBR interface.
- Establish IBGP between PE and ASBR in the same AS.
- Establish EBGP between ASBR devices and enable send-label.
- Obtain Kompella VPLS configuration information including VPLS instance descriptive information, RT value, VE ID, planned site number, VE ID deviation, and interface information from the network administrator.

Caution

When Option C (multihop MP-EBGP) is applied to realize inter-AC Kompella L2VPN applications, the next hop will be changed to itself by default when such information is sent to the peer EBGP if the MP-EBGP connection is set up by the route reflector between ASs to switch NLRI information of L2VPN. To realize the Kompella L2VPN through Option C solution, the **neighbor next-hop-unchanged** command must be configured on the route reflector so that the reflector does not change the next hop when NLRI information is sent. Otherwise, the inter-AS forwarding fails.

Configuration Steps

- Configuring CE1:

See "Configuring CE1" in basic configuration examples.

- Configuring PE1:

# Configure the loopback interface address.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 1.1.1.1 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 1.1.1.1 0.0.0.0 area 0
Ruijie(config-router)# network 10.10.10.0 0.0.0.255 area 0
Ruijie(config-router)# exit
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# exit
```

# Configure the public network tunnel between PEs.

```
Ruijie(config)# interface gigabitEthernet 1/2
Ruijie(config-if-Gigabitethernet 1/2)# ip address 10.10.10.1 255.255.255.0
Ruijie(config-if-Gigabitethernet 1/2)# mpls ip
Ruijie(config-if-Gigabitethernet 1/2)# label-switching
Ruijie(config-if-Gigabitethernet 1/2)# exit
```

# Configure the L2VPN address family.

```
Ruijie(config)# router bgp 100
Ruijie(config-router)# neighbor 4.4.4.4 remote-as 200
Ruijie(config-router)# neighbor 4.4.4.4 update-source loopback 0
Ruijie(config-router)# neighbor 4.4.4.4  ebgp-multihop
Ruijie(config-router)# address-family ipv4
Ruijie(config-router-af)# no neighbor 4.4.4.4  activate
Ruijie(config-router-af)# exit
Ruiije(config-router)# address-family l2vpn vpls
Ruijie(config-router-af)# neighbor 4.4.4.4 activate
Ruijie(config-router-af)# neighbor 4.4.4.4 send-community extended
Ruijie(config-router-af)# exit
```

# Configure a VLAN interface.

```
Ruijie(config)# vlan 10
Ruijie(config-vlan)# exit
```

# Configure the interface that connects CEs.

```
Ruijie(config)# interface gigabitEthernet 1/1
Ruijie(config-if-GigabitEthernet 1/1)# switchport access vlan 10
Ruijie(config-if-GigabitEthernet 1/1)# exit
```

# Configure a VPLS instance.

```
Ruijie# configure terminal
Ruijie(config)# l2 vfi vpls-1 vpnid 1 autodiscovery
Ruijie(config-vfi)# rd 100:1
Ruijie(config-vfi)# signal bgp
Ruijie(config-vfi)# encapsulation mpls ethernet
Ruijie(config-vfi)# route-target both 10000:1
Ruijie(config-vfi)# site-id 1
Ruijie(config-vfi-site)# xconnect interface vlan 10
Ruijie(config-vfi-site)#exit-site-mode
Ruijie(config-vfi)#exit
```

- Configuring ASBR1:

# Configure the loopback interface address.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 2.2.2.2 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# redistribute bgp subnets
Ruijie(config-router)# network 2.2.2.2 0.0.0.0 area 0
Ruijie(config-router)# network 10.10.10.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# advertise-labels for bgp-routes
Ruijie(config-mpls-router)# exit
```

#Configure the public network tunnel between PEs.

```
Ruijie(config)# interface gigabitEthernet 1/1
Ruijie(config-if-Gigabitethernet 1/1)# ip address 10.10.10.2 255.255.255.0
Ruijie(config-if-Gigabitethernet 1/1)# mpls ip
Ruijie(config-if-Gigabitethernet 1/1)# label-switching
Ruijie(config-if-Gigabitethernet 1/1)# exit
```

# Configure the interface that connects ASBRs.

```
Ruijie(config)# interface gigabitEthernet 1/2
Ruijie(config-if-Gigabitethernet 1/2)# ip address 192.168.1.1 255.255.255.252
Ruijie(config-if-Gigabitethernet 1/2)# label-switching
Ruijie(config-if-Gigabitethernet 1/2)# exit
```

# Enable ASBRs to allocate labels for PEs' routes.

```
Ruijie(config)# router bgp 100
Ruijie(config-router)# neighbor 192.168.1.2 remote-as 200
Ruijie(config-router)# address-family ipv4
Ruijie(config-router-af)# neighbor 192.168.1.2 send-label
Ruijie(config-router-af)# network 1.1.1.1 mask 255.255.255.255
Ruijie(config-router-af)# end
```

- Configuring ASBR2:

# Configure the loopback interface address.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 3.3.3.3 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes.

```
Ruijie(config)# router ospf 20
Ruijie(config-router)# redistribute bgp subnets
Ruijie(config-router)# network 3.3.3.3 0.0.0.0 area 0
Ruijie(config-router)# network 20.20.20.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# advertise-labels for bgp-routes
Ruijie(config-mpls-router)# exit
```

# Configure the public network tunnel between PEs.

```
Ruijie(config)# interface gigabitEthernet 1/2
Ruijie(config-if-Gigabitethernet 1/2)# ip address 20.20.20.2 255.255.255.0
Ruijie(config-if-Gigabitethernet 1/2)# mpls ip
Ruijie(config-if-Gigabitethernet 1/2)# label-switching
Ruijie(config-if-Gigabitethernet 1/2)# exit
```

# Configure the interface that connects ASBRs.

```
Ruijie(config)# interface gigabitEthernet 1/1
Ruijie(config-if-Gigabitethernet 1/1)# ip address 192.168.1.1 255.255.255.252
Ruijie(config-if-Gigabitethernet 1/1)# label-switching
Ruijie(config-if-Gigabitethernet 1/1)# exit
```

# Enable ASBRs to allocate labels for PEs' routes.

```
Ruijie(config)# router bgp 200
Ruijie(config-router)# neighbor 192.168.1.1 remote-as 100
Ruijie(config-router)# address-family ipv4
Ruijie(config-router-af)# neighbor 192.168.1.1  send-label
Ruijie(config-router-af)# network 4.4.4.4 mask 255.255.255.255
Ruijie(config-router-af)# end
```

- Configuring PE2:

# Configure the loopback interface address.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 4.4.4.4 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes.

```
Ruijie(config)# router ospf 20
Ruijie(config-router)# network 4.4.4.4 0.0.0.0 area 0
Ruijie(config-router)# network 20.20.20.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# exit
```

# Configure the public network tunnel between PEs.

```
Ruijie(config)# interface gigabitEthernet 1/1
Ruijie(config-if-Gigabitethernet 1/1)# ip address 20.20.20.1 255.255.255.0
Ruijie(config-if-Gigabitethernet 1/1)# mpls ip
Ruijie(config-if-Gigabitethernet 1/1)# label-switching
Ruijie(config-if-Gigabitethernet 1/1)# exit
```

# Configure the L2VPN address family.

```
Ruijie(config)# router bgp 100
Ruijie(config-router)# neighbor 1.1.1.1 remote-as 200
Ruijie(config-router)# neighbor 1.1.1.1 update-source loopback 0
Ruijie(config-router)# neighbor  1.1.1.1  ebgp-multihop
Ruijie(config-router)# address-family ipv4
Ruijie(config-router-af)# no neighbor  1.1.1.1 activate
Ruijie(config-router-af)# exit
Ruiije(config-router)# address-family l2vpn vpls
Ruijie(config-router-af)# neighbor 1.1.1.1 activate
Ruijie(config-router-af)# neighbor 1.1.1.1 send-community extended
Ruijie(config-router-af)# exit
```

# Configure a VLAN interface.

```
Ruijie(config)# vlan 10
Ruijie(config-vlan)# exit
```

# Enable the interface that connects PEs and CEs to bind the VPLS instance.

```
Ruijie(config)# interface gigabitEthernet 1/2
Ruijie(config-if-GigabitEthernet 1/2)# switchport access vlan 10
Ruijie(config-if-GigabitEthernet 1/2)# exit
```

# Configure a VPLS instance.

```
Ruijie# configure terminal
Ruijie(config)# l2 vfi vpls-1 vpnid 1 autodiscovery
Ruijie(config-vfi)# rd 200:1
Ruijie(config-vfi)# signal bgp
Ruijie(config-vfi)# encapsulation mpls ethernet
Ruijie(config-vfi)# route-target both 10000:1
Ruijie(config-vfi)# mtu 1500
Ruijie(config-vfi)# site-id 2
Ruijie(config-vfi-site)# xconnect interface vlan 10
Ruijie(config-vfi-site)#exit-site-mode
Ruijie(config-vfi)#exit
```

- Configuring CE2:

See "Configuring CE2" in basic configuration examples.

Verification

After the configuration, CE1 can ping with CE2.

After completing the configuration of Kompella VPLS, use the following commands to check the operation of VPLS.

Command	Function
Ruijie# **show bgp l2vpn vpls all**	Displays all the VPLS information.
Ruijie# **show mpls l2transport vc** [*vc_id* [*ip-address*]] | [**interface** *interface_name*] [**detail**]	Displays information about the PW (including VPWS PW and VPLS PW)
Ruijie# **show bgp l2vpn** { **vpls | vpws** } **all connection**s [**neighbor** *address*] | [**interface** *interface_name*] | [**site-id** *id*] [**detail**]	Displays VPLS PW information.
Ruijie# s**how mpls vfi** [*name*]	Displays all the configured or specified VFI information.

Typical Examples of Kompella VPLS Configuration for Routers

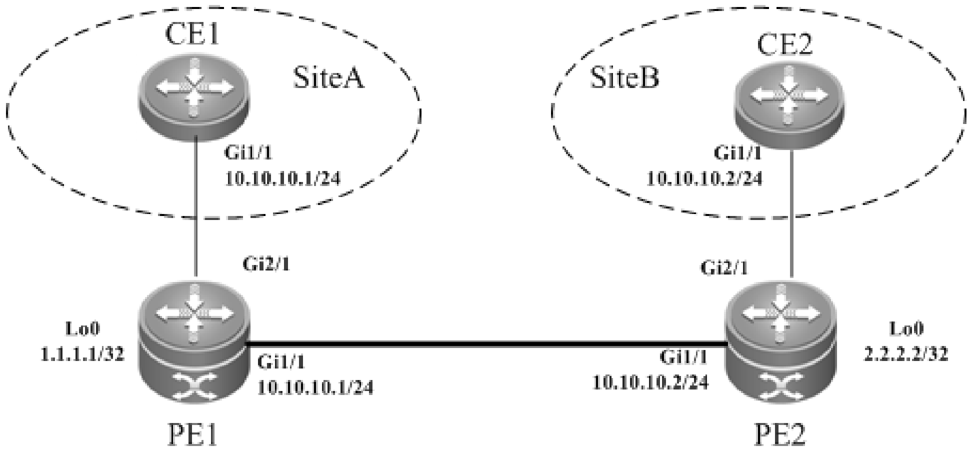
Basic Configuration Examples (Ethernet Access)

Networking Requirements

- LAN segments of customer S in site A and site B are connected with each other through the carrier's network devices, PE1 and PE2, forming a virtual and simulative LAN service or VPLS service.
- PE1 and PE2 are in the same AS and can automatically detect PE devices involved in the VPLS instance.
- Customer S's long-term network deployment plan is to connect LANs in five sites at most.
- The signaling protocol adopted by the carrier is the MP-BGP4 protocol.
- PEs are connected to CEs through the Ethernet interface.

Networking Topology

Figure 79 Basic Kompella VPLS configuration networking topology



Configuration Tips

Before configuring Kompella VPLS, complete the following tasks:

- Configure the public network tunnel that transmits data frames between VPLS PEs.
- Enable the L2VPN address family on PEs.
- Obtain Kompella VPLS configuration information including VPLS instance descriptive information, RT value, VE ID, planned site number, VE ID deviation, and interface information from the network administrator, and configure the Kompella VPLS instance.
- Configure the user access VPLS.

Configuration Steps

- Configuring CE1:

# Configure OSPF.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 10.10.10.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure the interface between CEs and PEs.

```
Ruijie(config)# interface gigabitEthernet 1/1
Ruijie(config-if-Gigabitethernet 1/1)# ip ref
Ruijie(config-if-Gigabitethernet 1/1)# ip address 10.10.10.1 255.255.255.0
Ruijie(config-router)# exit
```

- Configuring PE1

# Configure the loopback interface address.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 1.1.1.1 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 1.1.1.1 0.0.0.0 area 0
Ruijie(config-router)# network 10.10.10.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# exit
```

# Configure the public network tunnel between PEs.

```
Ruijie(config)# interface gigabitEthernet 1/1
Ruijie(config-if-Gigabitethernet 1/1)# ip ref
Ruijie(config-if-Gigabitethernet 1/1)# ip address 10.10.10.1 255.255.255.0
Ruijie(config-if-Gigabitethernet 1/1)# mpls ip
Ruijie(config-if-Gigabitethernet 1/1)# label-switching
Ruijie(config-if-Gigabitethernet 1/1)# exit
```

# Configure the L2VPN address family.

```
Ruijie(config)# router bgp 100
Ruijie(config-router)# neighbor 2.2.2.2 remote-as 100
Ruijie(config-router)# neighbor 2.2.2.2 update-source loopback 0
Ruiije(config-router)# address-family l2vpn vpls
Ruijie(config-router-af)# neighbor 2.2.2.2 activate
Ruijie(config-router-af)# neighbor 2.2.2.2 send-community extended
Ruijie(config-router-af)# exit
```

# Configure a VPLS instance.

```
Ruijie# configure terminal
Ruijie(config)# l2 vfi vpls-1 vpnid 1 autodiscovery
Ruijie(config-vfi)# rd 1:1
Ruijie(config-vfi)# signal bgp
Ruijie(config-vfi)# encapsulation mpls ethernet
Ruijie(config-vfi)# route-target both 10000:1
Ruijie(config-vfi)# site-id 1
Ruijie(config-vfi-site)# xconnect interface gigabitEthernet 2/1
Ruijie(config-vfi-site)#exit-site-id
Ruijie(config-vfi)# exit
```

# Configure the interface.

```
Ruijie(config)# interface gigabitEthernet 2/1
Ruijie(config-if-GigabitEthernet 2/1)# ip ref
Ruijie(config-if-GigabitEthernet 2/1)# exit
```

- Configuring PE2:

# Configure the loopback interface address.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 2.2.2.2 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 2.2.2.2 0.0.0.0 area 0
Ruijie(config-router)# network 10.10.10.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# exit
```

# Configure the public network tunnel between PEs.

```
Ruijie(config)# interface gigabitEthernet 1/1
Ruijie(config-if-Gigabitethernet 1/1)# ip ref
Ruijie(config-if-Gigabitethernet 1/1)# ip address 10.10.10.2 255.255.255.0
Ruijie(config-if-Gigabitethernet 1/1)# mpls ip
Ruijie(config-if-Gigabitethernet 1/1)# label-switching
Ruijie(config-if-Gigabitethernet 1/1)# exit
```

# Configure the L2VPN address family.

```
Ruijie(config)# router bgp 100
Ruijie(config-router)# neighbor 1.1.1.1 remote-as 100
Ruijie(config-router)# neighbor 1.1.1.1 update-source loopback 0
Ruiije(config-router)# address-family l2vpn vpls
Ruijie(config-router-af)# neighbor 1.1.1.1 activate
Ruijie(config-router-af)# neighbor 1.1.1.1 send-community extended
Ruijie(config-router-af)# exit
```

# Configure a VPLS instance.

```
Ruijie# configure terminal
Ruijie(config)# l2 vfi vpls-1 vpnid 1 autodiscovery
Ruijie(config-vfi)# rd 2:2
Ruijie(config-vfi)# signal bgp
Ruijie(config-vfi)# encapsulation mpls ethernet
Ruijie(config-vfi)# route-target both 10000:1
Ruijie(config-vfi)# site-id 2
Ruijie(config-vfi-site)# xconnect interface gigabitEthernet 2/1
Ruijie(config-vfi-site)#exit-site-id
Ruijie(config-vfi)#exit
```

# Configure the interface.

```
Ruijie(config)# interface gigabitEthernet 2/1
Ruijie(config-if-GigabitEthernet 2/1)# ip ref
Ruijie(config-if-GigabitEthernet 2/1)# exit
```

- Configuring CE2:

# Configure OSPF.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 10.10.10.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure the interface between CEs and PEs.

```
Ruijie(config)# interface gigabitEthernet 1/1
Ruijie(config-if-Gigabitethernet 1/1)# ip ref
Ruijie(config-if-Gigabitethernet 1/1)# ip address 10.10.10.2 255.255.255.0
Ruijie(config-router)# exit
```

Verification

After the configuration, CE1 can ping with CE2.

After completing the configuration of Kompella VPLS, use the following commands to check the operation of VPLS.

Command	Function
Ruijie# **show bgp l2vpn vpls all**	Displays all the VPLS information.
Ruijie# **show mpls l2transport vc** [*vc_id* [*ip-address*]] | [**interface** *interface_name*] [**detail**]	Displays information about PW (including VPWS PW and VPLS PW)
Ruijie# **show bgp l2vpn** { **vpls | vpws** } **all connections** [**neighbor** *address*] | [**interface** *interface_name*] | [**site-id** *id*] [**detail**]	Displays VPLS PW information.
Ruijie# **show mpls vfi** [*name*]	Displays all the configured or specified VFI information.

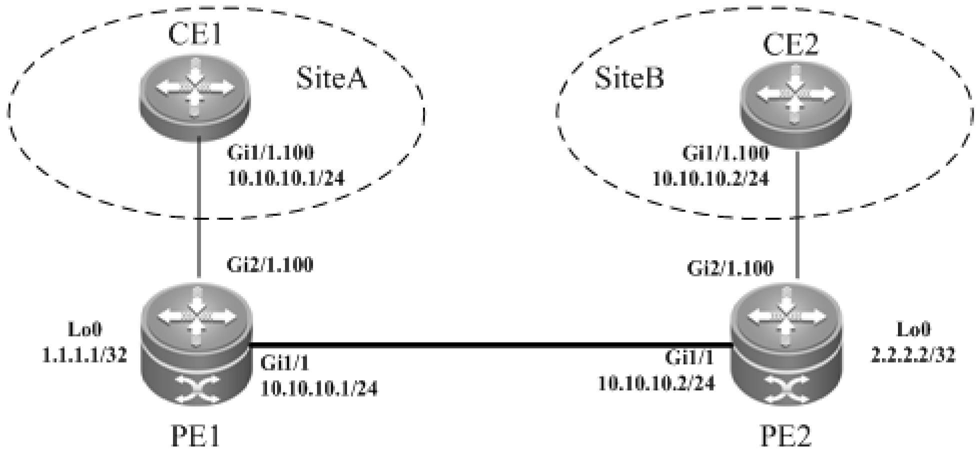
Basic Configuration Examples (Ethernet Sub-interface Access)

Networking Requirements

- LAN segments of customer S in site A and site B are connected with each other through the carrier's network devices, PE1 and PE2, forming a virtual and simulative LAN service or VPLS service.
- PE1 and PE2 are in the same AS and can automatically detect PE devices involved in the VPLS instance.
- Customer S's long-term network deployment plan is to connect LANs in five sites at most.
- The signaling protocol adopted by the carrier is the MP-BGP4 protocol.
- PEs are connected to CEs through the Ethernet sub-interface.

Networking Topology

Figure 80 Basic Kompella VPLS configuration networking topology



Configuration Tips

Before configuring Kompella VPLS, complete the following tasks:

- Configure the public network tunnel that transmits data frames between VPLS PEs.
- Enable the L2VPN address family on PEs.
- Obtain Kompella VPLS configuration information including VPLS instance descriptive information, RT value, VE ID, planned site number, VE ID deviation, and interface information from the network administrator, and configure the Kompella VPLS instance.
- Configure the user access VPLS.

Configuration Steps

- Configuring CE1:

# Configure OSPF.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 10.10.10.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure the interface between CEs and PEs.

```
Ruijie(config)# interface gigabitEthernet 1/1
Ruijie(config-if-Gigabitethernet 1/1)# ip ref
Ruijie(config-if-Gigabitethernet 1/1)# exit
Ruijie(config)# interface gigabitEthernet 1/1.100
Ruijie(config-if-Gigabitethernet 1/1.100)# encapsulation dot1Q 100
Ruijie(config-if-Gigabitethernet 1/1.100)# ip address 10.10.10.1 255.255.255.0
Ruijie(config-router)# exit
```

- Configuring PE1:

# Configure the loopback interface address.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 1.1.1.1 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 1.1.1.1 0.0.0.0 area 0
Ruijie(config-router)# network 10.10.10.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# exit
```

# Configure the public network tunnel between PEs.

```
Ruijie(config)# interface gigabitEthernet 1/1
Ruijie(config-if-Gigabitethernet 1/1)# ip ref
Ruijie(config-if-Gigabitethernet 1/1)# ip address 10.10.10.1 255.255.255.0
Ruijie(config-if-Gigabitethernet 1/1)# mpls ip
Ruijie(config-if-Gigabitethernet 1/1)# label-switching
Ruijie(config-if-Gigabitethernet 1/1)# exit
```

# Configure the L2VPN address family.

```
Ruijie(config)# router bgp 100
Ruijie(config-router)# neighbor 2.2.2.2 remote-as 100
Ruijie(config-router)# neighbor 2.2.2.2 update-source loopback 0
Ruiije(config-router)# address-family l2vpn vpls
Ruijie(config-router-af)# neighbor 2.2.2.2 activate
Ruijie(config-router-af)# neighbor 2.2.2.2 send-community extended
Ruijie(config-router-af)# exit
```

# Configure an interface.

```
Ruijie(config)# interface gigabitethernet 2/1
Ruijie(config-if-Gigabitethernet 2/1)# ip ref
Ruijie(config-if-Gigabitethernet 2/1)# exit
Ruijie(config)# interface gigabitEthernet 2/1.100
Ruijie(config-if-Gigabitethernet 2/1.100)# encapsulation dot1Q 100
Ruijie(config-if-GigabitEthernet 2/1.100)# exit
```

# Configure a VPLS instance.

```
Ruijie# configure terminal
Ruijie(config)# l2 vfi vpls-1 vpnid 1 autodiscovery
Ruijie(config-vfi)# rd 1:1
Ruijie(config-vfi)# signal bgp
Ruijie(config-vfi)# encapsulation mpls ethernetvlan
Ruijie(config-vfi)# route-target both 10000:1
Ruijie(config-vfi)# site-id 1
Ruijie(config-vfi-site)# xconnect interface gigabitEthernet 2/1.100
Ruijie(config-vfi-site)#exit-site-mode
Ruijie(config-vfi)#exit
```

- Configuring PE2:

# Configure the loopback interface address.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 2.2.2.2 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 2.2.2.2 0.0.0.0 area 0
Ruijie(config-router)# network 10.10.10.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# exit
```

# Configure the public network tunnel between PEs.

```
Ruijie(config)# interface gigabitEthernet 1/1
Ruijie(config-if-Gigabitethernet 1/1)# ip ref
Ruijie(config-if-Gigabitethernet 1/1)# ip address 10.10.10.2 255.255.255.0
Ruijie(config-if-Gigabitethernet 1/1)# mpls ip
Ruijie(config-if-Gigabitethernet 1/1)# label-switching
Ruijie(config-if-Gigabitethernet 1/1)# exit
```

# Configure the L2VPN address family.

```
Ruijie(config)# router bgp 100
Ruijie(config-router)# neighbor 1.1.1.1 remote-as 100
Ruijie(config-router)# neighbor 1.1.1.1 update-source loopback 0
Ruiije(config-router)# address-family l2vpn vpls
Ruijie(config-router-af)# neighbor 1.1.1.1 activate
Ruijie(config-router-af)# neighbor 1.1.1.1 send-community extended
Ruijie(config-router-af)# exit
```

# Bind the interface to the VPLS instance.

```
Ruijie(config)# interface gigabitEthernet 2/1
Ruijie(config-if-GigabitEthernet 2/1)# ip ref
Ruijie(config-if-GigabitEthernet 2/1)# exit
Ruijie(config)# interface gigabitEthernet 2/1.100
Ruijie(config-if-Gigabitethernet 2/1.100)# encapsulation dot1Q 100
Ruijie(config-if-GigabitEthernet 2/1.100)# exit
```

# Configure a VPLS instance.

```
Ruijie# configure terminal
Ruijie(config)# l2 vfi vpls-1 vpnid 1 autodiscovery
Ruijie(config-vfi)# rd 2:2
Ruijie(config-vfi)# signal bgp
Ruijie(config-vfi)# encapsulation mpls ethernetvlan
Ruijie(config-vfi)# route-target both 10000:1
Ruijie(config-vfi)# site-id 2
Ruijie(config-vfi-site)# xconnect interface gigabitEthernet 2/1.100
Ruijie(config-vfi-site)#exit-site-mode
Ruijie(config-vfi)#exit
```

- Configuring CE2:

# Configure OSPF.

```
Ruijie# configure terminal
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 10.10.10.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure the interface between CEs and PEs.

```
Ruijie(config)# interface gigabitEthernet 1/1.100
Ruijie(config-if-Gigabitethernet 1/1.100)# encapsulation dot1Q 100
Ruijie(config-if-Gigabitethernet 1/1.100)# ip ref
Ruijie(config-if-Gigabitethernet 1/1.100)# ip address 10.10.10.2 255.255.255.0
Ruijie(config-router)# exit
```

Verification

After the configuration, CE1 can ping with CE2.

After completing the configuration of Kompella VPLS, use the following commands to check the operation of VPLS.

Command	Function
Ruijie# **show bgp l2vpn vpls all**	Displays all the VPLS information.
Ruijie# **show mpls l2transport vc** [*vc_id* [*ip-address*]] | [**interface** *interface_name*] [**detail**]	Displays information about the PW (including VPWS PW and VPLS PW).
Ruijie# **show bgp l2vpn** { **vpls | vpws** } **all connections** [**neighbor** *address*] | [**interface** *interface_name*] | [**site-id** *id*] [**detail**]	Displays VPLS PW information.
Ruijie# s**how mpls vfi** [*name*]	Displays all the configured or specified VFI information.

Inter-AS Configuration Examples – Option A Solution

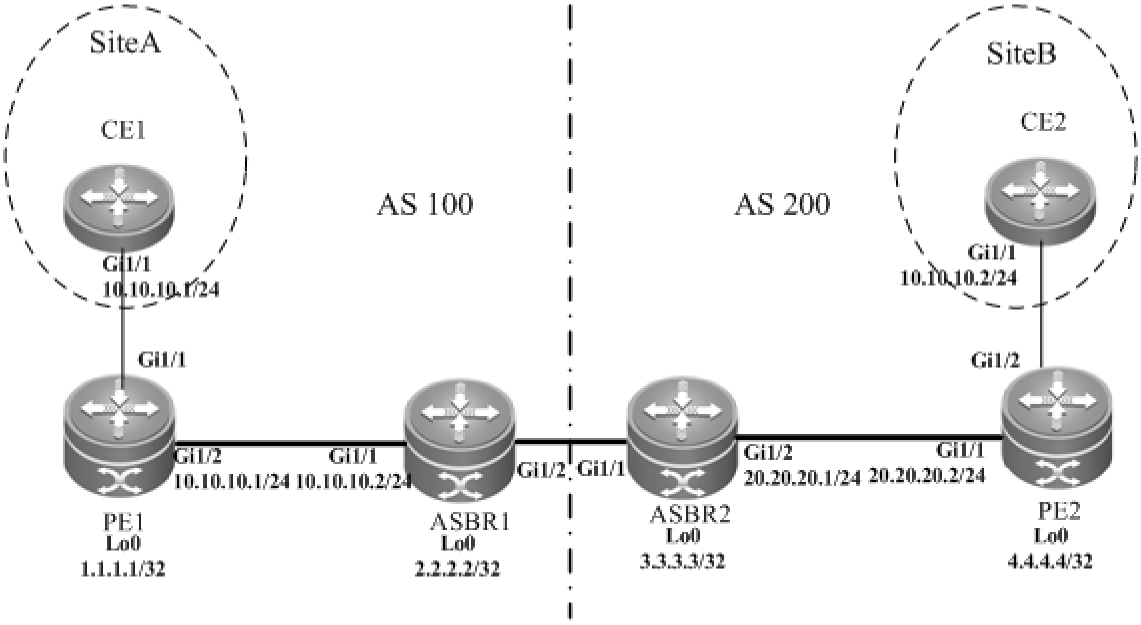
Networking Requirements

- LAN segments of customer S in site A and site B are connected with each other through the carrier's PE1 in AS 100 and PE2 in AS 200, forming a virtual and simulative LAN service or VPLS service;

- PE1 and PE2 are in different ASs. ASBR1 and ASBR2 are considered CE devices by each other, which means that the interface between ASBRs connects the AC to the VPLS instance.

Networking Topology

Figure 81 Kompella VPLS inter-AS networking topology



The preceding figure shows the structure of the Kompella VPLS inter-AS networking topology in Option A. The intermediate interface is considered by ASBRs as AC connection.

Configuration Tips

Before configuring Kompella VPLS, complete the following tasks:

- Run IGP in the carrier's network to realize connection between PE and ASBR devices.
- Establish the MP-IBGP peer relationship between PEs and intra-domain ASBRs in the domain.
- Obtain Kompella VPLS configuration information including VPLS instance descriptive information, RT value, VE ID, maximum planned site number, VE ID deviation, and interface information from the network administrator.

Configuration Steps

- Configuring CE1:

See "Configuring CE1" in basic configuration examples.

- Configuring PE1:

# Configure the loopback interface address.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 1.1.1.1 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes so that PEs can ping with ASBRs in the same AS.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 1.1.1.1 0.0.0.0 area 0
Ruijie(config-router)# network 10.10.10.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# exit
```

# Configure the public network tunnel between PEs.

```
Ruijie(config)# interface gigabitEthernet 1/2
Ruijie(config-if-Gigabitethernet 1/2)# ip ref
Ruijie(config-if-Gigabitethernet 1/2)# ip address 10.10.10.1 255.255.255.0
Ruijie(config-if-Gigabitethernet 1/2)# mpls ip
Ruijie(config-if-Gigabitethernet 1/2)# label-switching
Ruijie(config-if-Gigabitethernet 1/2)# exit
```

# Configure the L2VPN address family.

```
Ruijie(config)# router bgp 100
Ruijie(config-router)# neighbor 2.2.2.2 remote-as 100
Ruijie(config-router)# neighbor 2.2.2.2 update-source loopback 0
Ruiije(config-router)# address-family l2vpn vpls
Ruijie(config-router-af)# neighbor 2.2.2.2 activate
Ruijie(config-router-af)# neighbor 2.2.2.2 send-community extended
Ruijie(config-router-af)# exit
```

# Configure the interface between CEs and PEs.

```
Ruijie(config)# interface gigabitEthernet 1/1
Ruijie(config-if-GigabitEthernet 1/1)# ip ref
Ruijie(config-if-GigabitEthernet 1/1)# exit
```

# Configure a VPLS instance.

```
Ruijie# configure terminal
Ruijie(config)# l2 vfi vpls-1 vpnid 1 autodiscovery
Ruijie(config-vfi)# rd 100:1
Ruijie(config-vfi)# signal bgp
Ruijie(config-vfi)# encapsulation mpls ethernet
Ruijie(config-vfi)# route-target both 10000:1
Ruijie(config-vfi)# site-id 1
Ruijie(config-vfi-site)# xconnect interface gigabitEthernet 1/1
Ruijie(config-vfi-site)#exit-site-mode
```

- Configuring ASBR1:

# Configure the loopback interface address.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 2.2.2.2 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes so that PEs can ping with ASBRs in the same AS.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 2.2.2.2 0.0.0.0 area 0
Ruijie(config-router)# network 10.10.10.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# exit
```

# Configure the public network tunnel between PEs.

```
Ruijie(config)# interface gigabitEthernet 1/1
Ruijie(config-if-Gigabitethernet 1/1)# ip ref
Ruijie(config-if-Gigabitethernet 1/1)# ip address 10.10.10.2 255.255.255.0
Ruijie(config-if-Gigabitethernet 1/1)# mpls ip
Ruijie(config-if-Gigabitethernet 1/1)# label-switching
Ruijie(config-if-Gigabitethernet 1/1)# exit
```

# Configure the L2VPN address family.

```
Ruijie(config)# router bgp 100
Ruijie(config-router)# neighbor 1.1.1.1 remote-as 100
Ruijie(config-router)# neighbor 1.1.1.1 update-source loopback 0
Ruiije(config-router)# address-family l2vpn vpls
Ruijie(config-router-af)# neighbor 1.1.1.1 activate
Ruijie(config-router-af)# neighbor 1.1.1.1 send-community extended
Ruijie(config-router-af)# exit
```

# Configure the interface between ASBR1 and ASBR2.

```
Ruijie(config)# interface gigabitEthernet 1/2
Ruijie(config-if-GigabitEthernet 1/2)# ip ref
Ruijie(config-if-GigabitEthernet 1/2)# exit
```

# Configure a VPLS instance.

```
Ruijie# configure terminal
Ruijie(config)# l2 vfi vpls-1 vpnid 1 autodiscovery
Ruijie(config-vfi)# rd 100:1
Ruijie(config-vfi)# signal bgp
Ruijie(config-vfi)# encapsulation mpls ethernet
Ruijie(config-vfi)# route-target both 10000:1
Ruijie(config-vfi)# mtu 1500
Ruijie(config-vfi)# site-id 2
Ruijie(config-vfi-site)# xconnect interface gigabitEthernet 1/2
Ruijie(config-vfi-site)#exit-site-mode
Ruijie(config-vfi)#exit
```

- Configuring ASBR2:

# Configure the loopback interface address.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 3.3.3.3 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes so that PEs can ping with ASBRs in the same AS.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 3.3.3.3 0.0.0.0 area 0
Ruijie(config-router)# network 20.20.20.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# exit
```

# Configure the public network tunnel between PEs.

```
Ruijie(config)# interface gigabitEthernet 1/2
Ruijie(config-if-Gigabitethernet 1/2)# ip ref
Ruijie(config-if-Gigabitethernet 1/2)# ip address 20.20.20.2 255.255.255.0
Ruijie(config-if-Gigabitethernet 1/2)# mpls ip
Ruijie(config-if-Gigabitethernet 1/2)# label-switching
Ruijie(config-if-Gigabitethernet 1/2)# exit
```

# Configure the L2VPN address family.

```
Ruijie(config)# router bgp 200
Ruijie(config-router)# neighbor 4.4.4.4 remote-as 200
Ruijie(config-router)# neighbor 4.4.4.4 update-source loopback 0
Ruiije(config-router)# address-family l2vpn vpls
Ruijie(config-router-af)# neighbor 4.4.4.4 activate
Ruijie(config-router-af)# neighbor 4.4.4.4 send-community extended
Ruijie(config-router-af)# exit
```

# Configure the interface between ASBR1 and ASBR2.

```
Ruijie(config)# interface gigabitEthernet 1/1
Ruijie(config-if-GigabitEthernet 1/1)# ip ref
Ruijie(config-if-GigabitEthernet 1/1)# exit
```

# Configure a VPLS instance.

```
Ruijie# configure terminal
Ruijie(config)# l2 vfi vpls-1 vpnid 1 autodiscovery
Ruijie(config-vfi)# rd 200:1
Ruijie(config-vfi)# signal bgp
Ruijie(config-vfi)# encapsulation mpls ethernet
Ruijie(config-vfi)# route-target both 10000:1
Ruijie(config-vfi)# site-id 3
Ruijie(config-vfi-site)# xconnect interface gigabitEthernet 1/1
Ruijie(config-vfi-site)#exit-site-mode
Ruijie(config-vfi)#exit
```

- Configuring PE2:

# Configure the loopback interface address.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 4.4.4.4 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes so that PEs can ping with ASBRs in the same AS.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 4.4.4.4 0.0.0.0 area 0
Ruijie(config-router)# network 20.20.20.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# exit
```

# Configure the public network tunnel between PEs.

```
Ruijie(config)# interface gigabitEthernet 1/1
Ruijie(config-if-Gigabitethernet 1/1)# ip ref
Ruijie(config-if-Gigabitethernet 1/1)# ip address 20.20.20.1 255.255.255.0
Ruijie(config-if-Gigabitethernet 1/1)# mpls ip
Ruijie(config-if-Gigabitethernet 1/1)# label-switching
Ruijie(config-if-Gigabitethernet 1/1)# exit
```

# Configure the L2VPN address family.

```
Ruijie(config)# router bgp 200
Ruijie(config-router)# neighbor 3.3.3.3 remote-as 200
Ruijie(config-router)# neighbor 3.3.3.3 update-source loopback 0
Ruiije(config-router)# address-family l2vpn vpls
Ruijie(config-router-af)# neighbor 3.3.3.3 activate
Ruijie(config-router-af)# neighbor 3.3.3.3 send-community extended
Ruijie(config-router-af)# exit
```

# Configure the interface that connects PE2 and a CE.

```
Ruijie(config)# interface gigabitEthernet 1/2
Ruijie(config-if-GigabitEthernet 1/2)# ip ref
Ruijie(config-if-GigabitEthernet 1/2)# exit
```

# Configure a VPLS instance.

```
Ruijie# configure terminal
Ruijie(config)# l2 vfi vpls-1 vpnid 1 autodiscovery
Ruijie(config-vfi)# rd 200:1
Ruijie(config-vfi)# signal bgp
Ruijie(config-vfi)# encapsulation mpls ethernet
Ruijie(config-vfi)# route-target both 10000:1
Ruijie(config-vfi)# site-id 4
Ruijie(config-vfi-site)# xconnect interface gigabitEthernet 1/2
Ruijie(config-vfi-site)#exit-site-mode
Ruijie(config-vfi)#exit
```

- Configuring CE2:

See "Configuring CE2" in basic configuration examples.

Verification

After the configuration, CE1 can ping with CE2.

After completing the configuration of Kompella VPLS, use the following commands to check the operation of VPLS.

Command	**Function**
Ruijie# **show bgp l2vpn vpls all**	Displays all the VPLS information.
Ruijie# **show mpls l2transport vc** [*vc_id* [*ip-address*]] | [**interface** *interface_name*] [**detail**]	Displays information about PW (including VPWS PW and VPLS PW)
Ruijie# **show bgp l2vpn** { **vpls | vpws** } **all connections** [**neighbor** *address*] | [**interface** *interface_name*] | [**site-id** *id*] [**detail**]	Displays VPLS PW information.
Ruijie# **show mpls vfi** [*name*]	Displays all the configured or specified VFI information.

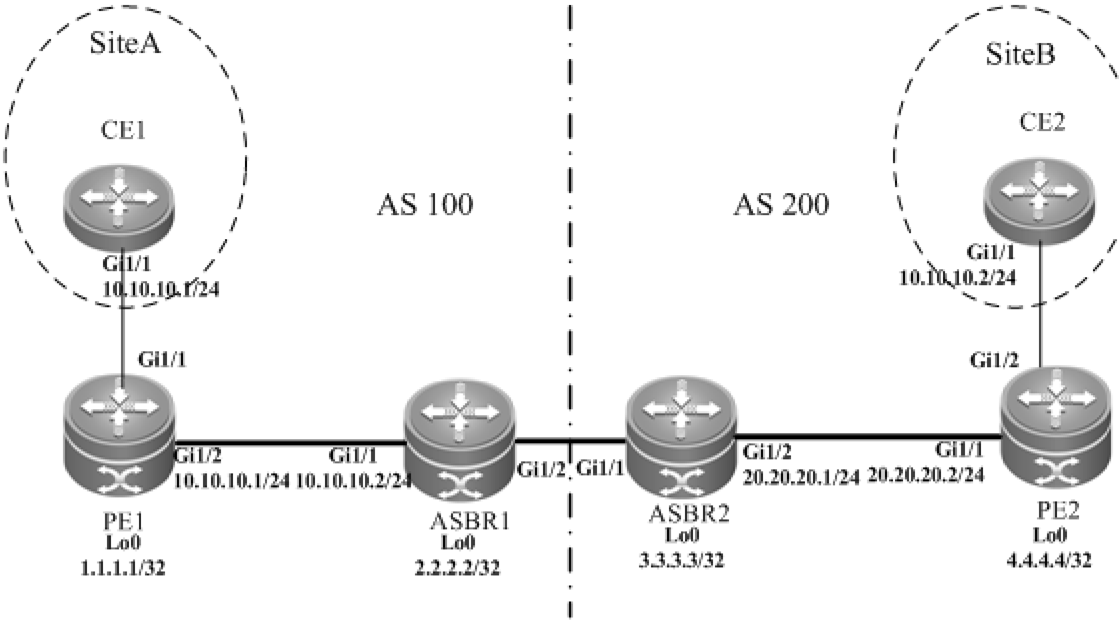
Inter-AS Configuration Examples – Option C Solution

Networking Requirements

- LAN segments of customer S in site A and site B are connected with each other through the carrier's PE1 in AS 1 and PE2 in AS 2, forming a virtual and simulative LAN service or VPLS service.
- PE1 and PE2 are in different ASs and can automatically detect PE devices involved in the VPLS instance.
- ASBR is not responsible for maintaining VPLS label block messages.
- VPLS label block messages are directly exchanged between PEs.

Networking Topology

Figure 50 Kompella VPLS Option C inter-AS networking topology



The preceding figure shows the structure of Kompella VPLS Option C inter-AS networking topology. Customer S's LAN segments in site A and site B are connected to each other through PE1 in AS 100 and PE2 in AS 200 as one LAN.

Configuration Tips

Before configuring Kompella VPLS, complete the following tasks:

- Run IGP in the carrier's network to realize connection between VPLS-PE and ASBR devices in the same AS.
- Establish a public network tunnel between PE and ASBR devices in the same AS and enable MPLS on the ASBR interface.
- Establish IBGP between PE and ASBR devices in the same AS.
- Establish EBGP between ASBR devices and enable send-label.
- Obtain Kompella VPLS configuration information including VPLS instance descriptive information, RT value, VE ID, planned site number, VE ID deviation, and interface information from the network administrator.

⚠ Caution

When Option C (multihop MP-EBGP) is applied to realize Inter-AC Kompella L2VPN applications, the next hop will be changed to itself by default when such information is sent to the peer EBGP if the MP-EBGP connection is set up by the route reflector between ASs to switch NLRI information of L2VPN. To realize the Kompella L2VPN through Option C solution, the **neighbor next-hop-unchanged** command must be configured on the route reflector so that the reflector does not change the next hop when NLRI information is sent. Otherwise, inter-AS forwarding fails.

Configuration Steps

- Configuring CE1:

See "Configuring CE1" in basic configuration examples.

- Configuring PE1:

# Configure the loopback interface address.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 1.1.1.1 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# network 1.1.1.1 0.0.0.0 area 0
Ruijie(config-router)# network 10.10.10.0 0.0.0.255 area 0
Ruijie(config-router)# exit
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# exit
```

# Configure the public network tunnel between PEs.

```
Ruijie(config)# interface gigabitEthernet 1/2
Ruijie(config-if-Gigabitethernet 1/2)# ip ref
Ruijie(config-if-Gigabitethernet 1/2)# ip address 10.10.10.1 255.255.255.0
Ruijie(config-if-Gigabitethernet 1/2)# mpls ip
Ruijie(config-if-Gigabitethernet 1/2)# label-switching
Ruijie(config-if-Gigabitethernet 1/2)# exit
```

# Configure the L2VPN address family.

```
Ruijie(config)# router bgp 100
Ruijie(config-router)# neighbor 4.4.4.4 remote-as 200
Ruijie(config-router)# neighbor 4.4.4.4 update-source loopback 0
Ruijie(config-router)# neighbor 4.4.4.4  ebgp-multihop
Ruijie(config-router)# address-family ipv4
Ruijie(config-router-af)# no neighbor 4.4.4.4  activate
Ruijie(config-router-af)# exit
Ruiije(config-router)# address-family l2vpn vpls
Ruijie(config-router-af)# neighbor 4.4.4.4 activate
Ruijie(config-router-af)# neighbor 4.4.4.4 send-community extended
Ruijie(config-router-af)# exit
```

# Configure the interface that connects CEs.

```
Ruijie(config)# interface gigabitEthernet 1/1
Ruijie(config-if-GigabitEthernet 1/1)# ip ref
Ruijie(config-if-GigabitEthernet 1/1)# exit
```

# Configure a VPLS instance.

```
Ruijie# configure terminal
Ruijie(config)# l2 vfi vpls-1 vpnid 1 autodiscovery
Ruijie(config-vfi)# rd 100:1
Ruijie(config-vfi)# signal bgp
Ruijie(config-vfi)# encapsulation mpls ethernet
Ruijie(config-vfi)# route-target both 10000:1
Ruijie(config-vfi)# site-id 1
Ruijie(config-vfi-site)# xconnect interface gigabitEthernet 1/1
Ruijie(config-vfi-site)#exit-site-mode
Ruijie(config-vfi)#exit
```

- Configuring ASBR1:

# Configure the loopback interface address.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 2.2.2.2 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes.

```
Ruijie(config)# router ospf 10
Ruijie(config-router)# redistribute bgp subnets
Ruijie(config-router)# network 2.2.2.2 0.0.0.0 area 0
Ruijie(config-router)# network 10.10.10.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# advertise-labels for bgp-routes
Ruijie(config-mpls-router)# exit
```

#Configure the public network tunnel between PEs.

```
Ruijie(config)# interface gigabitEthernet 1/1
Ruijie(config-if-Gigabitethernet 1/1)# ip ref
Ruijie(config-if-Gigabitethernet 1/1)# ip address 10.10.10.2 255.255.255.0
Ruijie(config-if-Gigabitethernet 1/1)# mpls ip
Ruijie(config-if-Gigabitethernet 1/1)# label-switching
Ruijie(config-if-Gigabitethernet 1/1)# exit
```

# Configure the interface that connects ASBRs.

```
Ruijie(config)# interface gigabitEthernet 1/2
Ruijie(config-if-Gigabitethernet 1/2)# ip ref
Ruijie(config-if-Gigabitethernet 1/2)# ip address 192.168.1.1 255.255.255.252
Ruijie(config-if-Gigabitethernet 1/2)# label-switching
Ruijie(config-if-Gigabitethernet 1/2)# exit
```

# Enable ASBRs to allocate labels for PEs' routes.

```
Ruijie(config)# router bgp 100
Ruijie(config-router)# neighbor 192.168.1.2 remote-as 200
Ruijie(config-router)# address-family ipv4
Ruijie(config-router-af)# neighbor 192.168.1.2 send-label
Ruijie(config-router-af)# network 1.1.1.1 mask 255.255.255.255
Ruijie(config-router-af)# end
```

- Configuring ASBR2:

# Configure the loopback interface address.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 3.3.3.3 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes.

```
Ruijie(config)# router ospf 20
Ruijie(config-router)# redistribute bgp subnets
Ruijie(config-router)# network 3.3.3.3 0.0.0.0 area 0
Ruijie(config-router)# network 20.20.20.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# advertise-labels for bgp-routes
Ruijie(config-mpls-router)# exit
```

# Configure the public network tunnel between PEs.

```
Ruijie(config)# interface gigabitEthernet 1/2
Ruijie(config-if-Gigabitethernet 1/2)# ip ref
Ruijie(config-if-Gigabitethernet 1/2)# ip address 20.20.20.2 255.255.255.0
Ruijie(config-if-Gigabitethernet 1/2)# mpls ip
Ruijie(config-if-Gigabitethernet 1/2)# label-switching
Ruijie(config-if-Gigabitethernet 1/2)# exit
```

# Configure the interface that connects ASBRs.

```
Ruijie(config)# interface gigabitEthernet 1/1
Ruijie(config-if-Gigabitethernet 1/1)# ip ref
Ruijie(config-if-Gigabitethernet 1/1)# ip address 192.168.1.1 255.255.255.252
Ruijie(config-if-Gigabitethernet 1/1)# label-switching
Ruijie(config-if-Gigabitethernet 1/1)# exit
```

# Enable ASBRs to allocate labels for PEs' routes.

```
Ruijie(config)# router bgp 200
Ruijie(config-router)# neighbor 192.168.1.1 remote-as 100
Ruijie(config-router)# address-family ipv4
Ruijie(config-router-af)# neighbor 192.168.1.1  send-label
Ruijie(config-router-af)# network 4.4.4.4 mask 255.255.255.255
Ruijie(config-router-af)# end
```

- Configuring PE2:

# Configure the loopback interface address.

```
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 4.4.4.4 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

# Configure OSPF and establish public network routes.

```
Ruijie(config)# router ospf 20
Ruijie(config-router)# network 4.4.4.4 0.0.0.0 area 0
Ruijie(config-router)# network 20.20.20.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

# Configure LDP and globally enable MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)# exit
```

# Configure the public network tunnel between PEs.

```
Ruijie(config)# interface gigabitEthernet 1/1
Ruijie(config-if-Gigabitethernet 1/1)# ip ref
Ruijie(config-if-Gigabitethernet 1/1)# ip address 20.20.20.1 255.255.255.0
Ruijie(config-if-Gigabitethernet 1/1)# mpls ip
Ruijie(config-if-Gigabitethernet 1/1)# label-switching
Ruijie(config-if-Gigabitethernet 1/1)# exit
```

# Configure the L2VPN address family.

```
Ruijie(config)# router bgp 100
Ruijie(config-router)# neighbor 1.1.1.1 remote-as 200
Ruijie(config-router)# neighbor 1.1.1.1 update-source loopback 0
Ruijie(config-router)# neighbor  1.1.1.1  ebgp-multihop
Ruijie(config-router)# address-family ipv4
Ruijie(config-router-af)# no neighbor  1.1.1.1 activate
Ruijie(config-router-af)# exit
Ruiije(config-router)# address-family l2vpn vpls
Ruijie(config-router-af)# neighbor 1.1.1.1 activate
Ruijie(config-router-af)# neighbor 1.1.1.1 send-community extended
Ruijie(config-router-af)# exit
```

# Enable the interface that connects PEs and CEs to bind the VPLS instance.

```
Ruijie(config)# interface gigabitEthernet 1/2
Ruijie(config-if-GigabitEthernet 1/2)# ip ref
Ruijie(config-if-GigabitEthernet 1/2)# exit
```

# Configure a VPLS instance.

```
Ruijie# configure terminal
Ruijie(config)# l2 vfi vpls-1 vpnid 1 autodiscovery
Ruijie(config-vfi)# rd 200:1
Ruijie(config-vfi)# signal bgp
Ruijie(config-vfi)# encapsulation mpls ethernet
Ruijie(config-vfi)# route-target both 10000:1
Ruijie(config-vfi)# mtu 1500
Ruijie(config-vfi)# site-id 2
Ruijie(config-vfi-site)# xconnect interface gigabitEthernet 1/2
Ruijie(config-vfi-site)#exit-site-mode
Ruijie(config-vfi)#exit
```

- Configuring CE2:

See "Configuring CE2" in basic configuration examples.

Verification

After the configuration, CE1 can ping with CE2.

After completing the configuration of Kompella VPLS, use the following commands to check the operation of VPLS.

Command	Function
Ruijie# **show bgp l2vpn vpls all**	Displays all the VPLS information.
Ruijie# **show mpls l2transport vc** [*vc_id* [*ip-address*]] | [**interface** *interface_name*] [**detail**]	Displays information about the PW (including VPWS PW and VPLS PW)
Ruijie# **show bgp l2vpn** { **vpls | vpws** } **all connections** [**neighbor** *address*] | [**interface** *interface_name*] | [**site-id** *id*] [**detail**]	Displays VPLS PW information.
Ruijie# **show mpls vfi** [*name*]	Displays all the configured or specified VFI information.

Examples of Maintaining PW Connectivity of VPLS

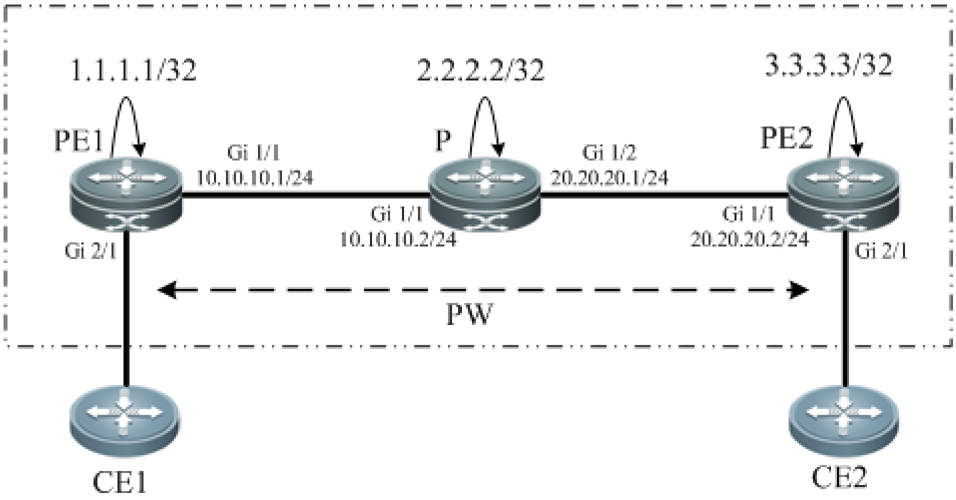
Verifying PW Connectivity and Fault of Martini VPLS

Networking Requirements

- The CE devices play the role of the customer network to connect to the PE devices via the AC.
- PE1 and PE2 establish a PW between them via LDP.
- PE1, P, and PE2 set up the public tunnel and LSP path via LDP.

Network Topology

Figure 51



Configuration Key Points

The configuration key points for PE1, P, and PE2 are as follows:

- Configure the IP address of the interface of each device and configure the OSPF protocol.
- Configure the global MPLS capability and the interface MPLS capability of each device.
- Configure LDP to enable the network to forward MPLS traffic.
- Configure the VPLS.
- Check the PW connectivity.
- Locate the PW fault.

Configuration Steps

(1) Configure the IP address of the interface of each node and configure the OSPF protocol.

#Configure PE1.

```
Ruijie(config)#interface loopback 0
Ruijie(config-if-Loopback 0)#ip address 1.1.1.1 255.255.255.255
Ruijie(config-if-Loopback 0)#exit
Ruijie(config-if-GigabitEthernet 1/1)#ip address 10.10.10.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/1)#exit
Ruijie(config)#router ospf 10
Ruijie(config-router)#network 1.1.1.1 0.0.0.0 area 0
Ruijie(config-router)#network 10.10.10.0 0.0.0.255 area 0
Ruijie(config-router)#exit
```

#Configure P.

```
Ruijie(config)#interface loopback 0
Ruijie(config-if-Loopback 0)#ip address 2.2.2.2 255.255.255.255
Ruijie(config-if-Loopback 0)#exit
Ruijie(config)#interface gigabitEthernet 1/1
Ruijie(config-if-GigabitEthernet 1/1)#ip address 10.10.10.2 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/1)#exit
Ruijie(config)#interface gigabitEthernet 1/2
Ruijie(config-if-GigabitEthernet 1/2)#ip address 20.20.20.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/2)#exit
Ruijie(config)#router ospf 10
Ruijie(config-router)#network 2.2.2.2 0.0.0.0 area 0
Ruijie(config-router)#network 10.10.10.0 0.0.0.255 area 0
Ruijie(config-router)#network 20.20.20.0 0.0.0.255 area 0
Ruijie(config-router)#exit
```

#Configure PE2.

```
Ruijie(config)#interface loopback 0
Ruijie(config-if-Loopback 0)#ip address 3.3.3.3 255.255.255.255
Ruijie(config-if-Loopback 0)#exit
Ruijie(config)#interface gigabitEthernet 1/1
Ruijie(config-if-GigabitEthernet 1/1)#ip address 20.20.20.2 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/1)#exit
Ruijie(config)#router ospf 10
Ruijie(config-router)#network 3.3.3.3 0.0.0.0 area 0
Ruijie(config-router)#network 20.20.20.0 0.0.0.255 area 0
Ruijie(config-router)#exit
```

(2) Configure the global MPLS capability and the interface MPLS capability of each node and enable the LDP function on the interfaces.

#Configure PE1.

```
Ruijie(config)#mpls ip
Ruijie(config)#mpls router ldp
Ruijie(config-mpls-router)#ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)#neighbor 3.3.3.3
Ruijie(config-mpls-router)#exit
Ruijie(config)#interface gigabitEthernet 1/1
Ruijie(config-if-GigabitEthernet 1/1)#mpls ip
Ruijie(config-if-GigabitEthernet 1/1)#label-switching
Ruijie(config-if-GigabitEthernet 1/1)#exit
```

#Configure P.

```
Ruijie(config)#mpls ip
Ruijie(config)#mpls router ldp
Ruijie(config-mpls-router)#ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)#exit
Ruijie(config)#interface gigabitEthernet 1/1
Ruijie(config-if-GigabitEthernet 1/1)#mpls ip
Ruijie(config-if-GigabitEthernet 1/1)#label-switching
Ruijie(config-if-GigabitEthernet 1/1)#exit
Ruijie(config)#interface gigabitethernet 1/2
Ruijie(config-if-GigabitEthernet 1/2)#mpls ip
Ruijie(config-if-GigabitEthernet 1/2)#label-switching
Ruijie(config-if-GigabitEthernet 1/2)#exit
```

#Configure PE2.

```
Ruijie(config)#mpls ip
Ruijie(config)#mpls router ldp
Ruijie(config-mpls-router)#ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)#neighbor 1.1.1.1
Ruijie(config-mpls-router)#exit
Ruijie(config)#interface gigabitEthernet 1/1
Ruijie(config-if-GigabitEthernet 1/1)#mpls ip
Ruijie(config-if-GigabitEthernet 1/1)#label-switching
Ruijie(config-if-GigabitEthernet 1/1)#exit
```

(3) Configure VPLS instances between PE1 and PE2 and enable CE1 and CE2 to access the VPLS.

#Configure PE1.

```
Ruijie(config)# l2 vfi vfi-a vpnid 1
Ruijie(config-vfi)# neighbor 1.1.1.1 vc-id 100 encapsulation mpls
Ruijie(config-vfi)# exit
Ruijie(config)# vlan 10
Ruijie(config-vlan)# exit
Ruijie(config)# interface vlan 10
Ruijie(config-if-Vlan 10)# xconnect vfi vfi-a
Ruijie(config-if-Vlan 10)# exit
Ruijie(config)# interface gigabitEthernet 3/2
Ruijie(config-if-Gigabitethernet 3/2)#  switchport access vlan 10
Ruijie(config-if-Gigabitethernet 3/2)# exit
```

#Configure PE2.

```
Ruijie(config)# l2 vfi vfi-a vpnid 1
Ruijie(config-vfi)# neighbor 3.3.3.3 vc-id 100 encapsulation mpls
Ruijie(config-vfi)# exit
Ruijie(config)# vlan 10
Ruijie(config-vlan)# exit
Ruijie(config)# interface vlan 10
Ruijie(config-if-Vlan 10)# xconnect vfi vfi-a
Ruijie(config-if-Vlan 10)# exit
Ruijie(config)# interface gigabitEthernet 1/3
Ruijie(config-if-Gigabitethernet 1/3)#  switchport access vlan 10
Ruijie(config-if-Gigabitethernet 1/3)# exit
```

(4) Verify the PW connectivity.

```
Ruijie#ping mpls pseudowire 3.3.3.3 100 label-alert
Sending 5, 112-byte MPLS Echoes on Pseudowire, peer address 3.3.3.3, VC ID 100,
    timeout is 2 seconds, send interval is 0 msec:

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
 'L'-labeled output interface,'B'-unlabeled output interface,
 'D'-DS Map mismatch,'F'-no FEC mapping,'f'-FEC mismatch,
 'M'-malformed request,'m'-unsupported tlvs,'N'-no label entry,
 'P'-no rx intf label prot,'p'-premature termination of LSP,
 'R'-transit router,'I'-unknown upstream index,
 'X'-unknown return code,'x'-return code 0
Press Ctrl+C to break.
!!!!!
Success rate is 100 percent(5/5),round-trip min/avg/max=20/36/60 m
```

(5) Locate the PW fault.

```
Ruijie#traceroute mpls pseudowire 3.3.3.3 100 segment 1 ttl-expiry
Traceing Pseudowire Label Switched Path, peer address 3.3.3.3, VC ID 100, timeout is 2
seconds

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Press Ctrl+C to break.
  0 10.10.10.2     MRU 1500 [Labels: 1025 Exp: 0]
! 1 20.20.20.2     60 ms [Labels: 1025 Exp: 0]
                local 1.1.1.1 remote 3.3.3.3 vc id 100
```

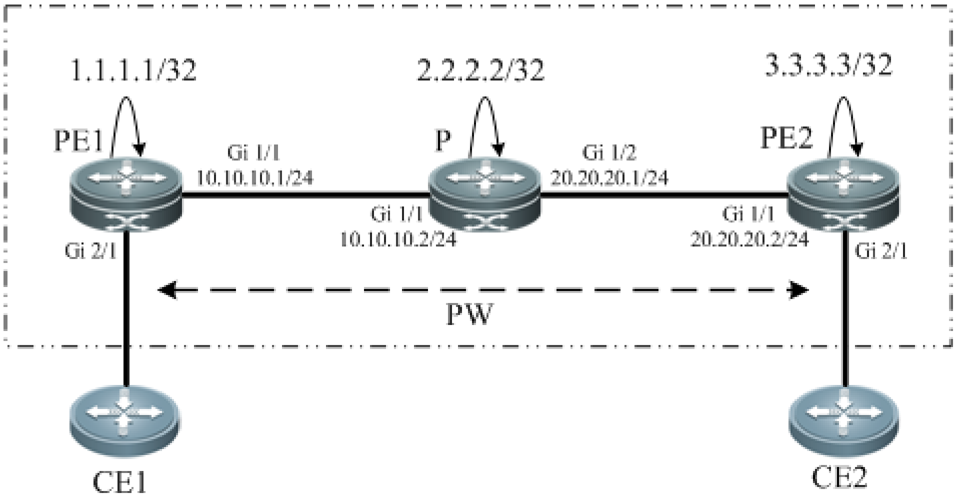
Verifying PW Connectivity of Kompella VPLS

Networking Requirements

As shown in Figure 3-36, PE1, P, and PE2 belong to the same MPLS domain, and CE1 and CE2 belong to the same VPN instance. PE1 and PE2 establish a PW between them and use the ping or traceroute command to check the PW connectivity to ensure services on the links.

Network Topology

Figure 52 错误!未指定顺序。



Configuration Key Points

The configuration key points for each node are as follows:

- Configure the IP address of the interface of each node and configure the OSPF protocol.
- Configure the global MPLS capability and the interface MPLS capability of each node.
- Enable the L2VPN on PE1 and PE2 and create a VC connection.
- Check the PW connectivity.
- Locate the PW fault.

Configuration Steps

(1) Configure the IP address of the interface of each node and configure the OSPF protocol.

#Configure PE1.

```
Ruijie(config)#interface loopback 0
Ruijie(config-if-Loopback 0)#ip address 1.1.1.1 255.255.255.255
Ruijie(config-if-Loopback 0)#exit
Ruijie(config-if-GigabitEthernet 1/1)#ip address 10.10.10.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/1)#exit
Ruijie(config)#router ospf 10
Ruijie(config-router)#network 1.1.1.1 0.0.0.0 area 0
Ruijie(config-router)#network 10.10.10.0 0.0.0.255 area 0
Ruijie(config-router)#exit
```

#Configure P.

```
Ruijie(config)#interface loopback 0
Ruijie(config-if-Loopback 0)#ip address 2.2.2.2 255.255.255.255
Ruijie(config-if-Loopback 0)#exit
Ruijie(config)#interface gigabitEthernet 1/1
Ruijie(config-if-GigabitEthernet 1/1)#ip address 10.10.10.2 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/1)#exit
Ruijie(config)#interface gigabitEthernet 1/2
Ruijie(config-if-GigabitEthernet 1/2)#ip address 20.20.20.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/2)#exit
Ruijie(config)#router ospf 10
Ruijie(config-router)#network 2.2.2.2 0.0.0.0 area 0
Ruijie(config-router)#network 10.10.10.0 0.0.0.255 area 0
Ruijie(config-router)#network 20.20.20.0 0.0.0.255 area 0
Ruijie(config-router)#exit
```

#Configure PE2.

```
Ruijie(config)#interface loopback 0
Ruijie(config-if-Loopback 0)#ip address 3.3.3.3 255.255.255.255
Ruijie(config-if-Loopback 0)#exit
Ruijie(config)#interface gigabitEthernet 1/1
Ruijie(config-if-GigabitEthernet 1/1)#ip address 20.20.20.2 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/1)#exit
Ruijie(config)#router ospf 10
Ruijie(config-router)#network 3.3.3.3 0.0.0.0 area 0
Ruijie(config-router)#network 20.20.20.0 0.0.0.255 area 0
Ruijie(config-router)#exit
```

(2) Configure the global MPLS capability and the interface MPLS capability of each node and enable the LDP function on the interfaces.

#Configure PE1.

```
Ruijie(config)#mpls ip
Ruijie(config)#mpls router ldp
Ruijie(config-mpls-router)#ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)#exit
Ruijie(config)#interface gigabitEthernet 1/1
Ruijie(config-if-GigabitEthernet 1/1)#mpls ip
Ruijie(config-if-GigabitEthernet 1/1)#label-switching
Ruijie(config-if-GigabitEthernet 1/1)#exit
```

#Configure P.

```
Ruijie(config)#mpls ip
Ruijie(config)#mpls router ldp
Ruijie(config-mpls-router)#ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)#exit
Ruijie(config)#interface gigabitEthernet 1/1
Ruijie(config-if-GigabitEthernet 1/1)#mpls ip
Ruijie(config-if-GigabitEthernet 1/1)#label-switching
Ruijie(config-if-GigabitEthernet 1/1)#exit
Ruijie(config)#interface gigabitEthernet 1/2
Ruijie(config-if-GigabitEthernet 1/2)#mpls ip
Ruijie(config-if-GigabitEthernet 1/2)#label-switching
Ruijie(config-if-GigabitEthernet 1/2)#exit
```

#Configure PE2.

```
Ruijie(config)#mpls ip
Ruijie(config)#mpls router ldp
Ruijie(config-mpls-router)#ldp router-id interface loopback 0  force
Ruijie(config-mpls-router)#exit
Ruijie(config)#interface gigabitEthernet 1/1
Ruijie(config-if-GigabitEthernet 1/1)#mpls ip
Ruijie(config-if-GigabitEthernet 1/1)#label-switching
Ruijie(config-if-GigabitEthernet 1/1)#exit
```

(3) Configure a L2VPN address family.

#Configure PE1.

```
Ruijie(config)#router bgp 100
Ruijie(config-router)#neighbor 3.3.3.3 remote-as 100
Ruijie(config-router)#neighbor 3.3.3.3 update-source loopback 0
Ruiije(config-router)#address-family l2vpn vpws
Ruijie(config-router-af)#neighbor 3.3.3.3 activate
Ruijie(config-router-af)#neighbor 3.3.3.3 send-community extended
Ruijie(config-router-af)#exit
```

#Configure PE2.

```
Ruijie(config)#router bgp 100
Ruijie(config-router)#neighbor 1.1.1.1 remote-as 100
Ruijie(config-router)#neighbor 1.1.1.1 update-source loopback 0
Ruiije(config-router)#address-family l2vpn vpws
Ruijie(config-router-af)#neighbor 1.1.1.1 activate
Ruijie(config-router-af)#neighbor 1.1.1.1 send-community extended
Ruijie(config-router-af)#exit
```

(4) Configure a VFI instance.

#Configure PE1.

```
Ruijie(config)# vlan 10
Ruijie(config-vlan)# exit
Ruijie(config)# interface gigabitEthernet 3/2
Ruijie(config-if-Gigabitethernet 3/2)#  switchport access vlan 10
Ruijie(config-if-Gigabitethernet 3/2)# exit
Ruijie(config)# l2 vfi vpls-1 vpnid 1 autodiscovery
Ruijie(config-vfi)# rd 1:1
Ruijie(config-vfi)# signal bgp
Ruijie(config-vfi)# encapsulation mpls ethernet
Ruijie(config-vfi)# route-target both 10000:1
Ruijie(config-vfi)# site-id 1
Ruijie(config-vfi-site)# xconnect interface vlan 10
Ruijie(config-vfi-site)#exit-site-mode
Ruijie(config-vfi)# eixt
```

#Configure PE2.

```
Ruijie(config)# vlan 10
Ruijie(config-vlan)# exit
Ruijie(config)# interface gigabitEthernet 1/3
Ruijie(config-if-Gigabitethernet 1/3)#  switchport access vlan 10
Ruijie(config-if-Gigabitethernet 1/3)# exit
Ruijie(config)# l2 vfi vpls-1 vpnid 1 autodiscovery
Ruijie(config-vfi)# rd 1:1
Ruijie(config-vfi)# signal bgp
Ruijie(config-vfi)# encapsulation mpls ethernet
Ruijie(config-vfi)# route-target both 10000:1
Ruijie(config-vfi)# site-id 1
Ruijie(config-vfi-site)# xconnect interface vlan 10
Ruijie(config-vfi-site)#exit-site-mode
Ruijie(config-vfi)# eixt
```

(5) Verify the PW connectivity.

```
Ruijie#ping mpls pseudowire kompella vpls-1 1 2 rd 1:1 label-alert
Sending 5, 112-byte MPLS Echoes on Pseudowire, VFI name vpls-1, local site id, 1, remote
site id 2,
    timeout is 2 seconds, send interval is 0 msec:

Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
 'L'-labeled output interface,'B'-unlabeled output interface,
 'D'-DS Map mismatch,'F'-no FEC mapping,'f'-FEC mismatch,
 'M'-malformed request,'m'-unsupported tlvs,'N'-no label entry,
 'P'-no rx intf label prot,'p'-premature termination of LSP,
 'R'-transit router,'I'-unknown upstream index,
 'X'-unknown return code,'x'-return code 0
Press Ctrl+C to break.
!!!!!
Success rate is 100 percent(5/5),round-trip min/avg/max=20/36/60 ms
```

(6) Locate the PW fault.

```
Ruijie#traceroute mpls pseudowire kompella vpls-1 1 2 rd 1:1 label-alert
```

```
Traceing Pseudowire Label Switched Path, VFI name vpls-1, local site id 1, remote site
id 2, timeout is 2 seconds.
Codes: '!' - success, 'Q' - request not sent, '.' - timeout,
  'L' - labeled output interface, 'B' - unlabeled output interface,
  'D' - DS Map mismatch, 'F' - no FEC mapping, 'f' - FEC mismatch,
  'M' - malformed request, 'm' - unsupported tlvs, 'N' - no label entry,
  'P' - no rx intf label prot, 'p' - premature termination of LSP,
  'R' - transit router, 'I' - unknown upstream index,
  'X' - unknown return code, 'x' - return code 0

Press Ctrl+C to break.
  0 10.10.10.2    MRU 1500 [Labels: router-alert/1538 Exp: 0/0]
! 1 20.20.20.2    20 ms
```

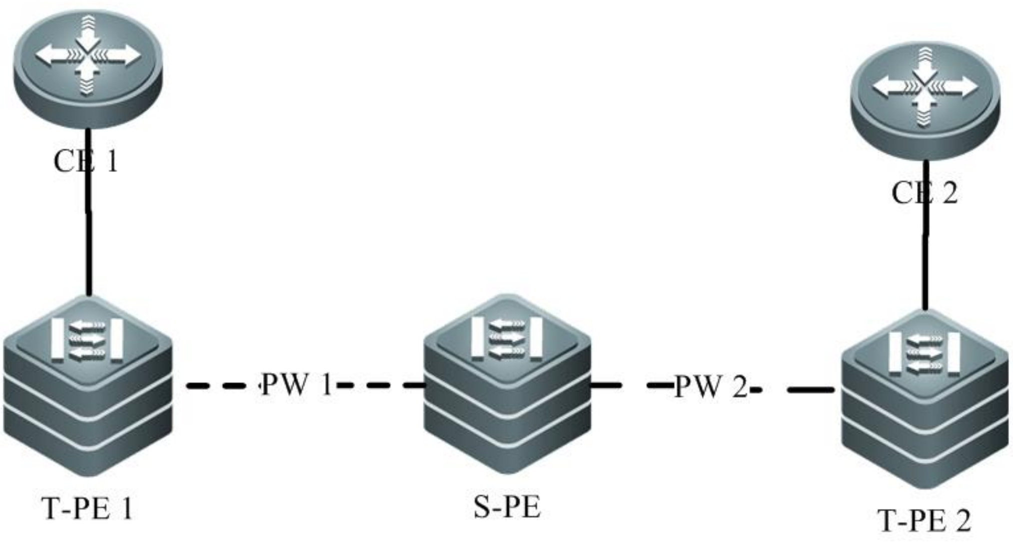
MS-PW

Introduction

MS-PW is an intra-domain or cross-domain end-to-end pseudo wire in the IP/MPLS network, and is set up between two user-oriented T-PE devices of the operator and passes through one or more PE devices (switches). It is used to simulate Layer 2 point-to-point services. The MS-PW allows you to carry the specific services from the AC across domains on the paths or tunnels between the ingress and egress of the PW and to manage the information such as the service-related signaling, statuses, and alarms on the border of the PW.

Overview

Figure 53 Basic MS-PW Topology



Involved devices are as follows:

- CE (customer edge): the device directly connected to the user network and operator network. It can be a router, a bridge device, or a host. The customer edge is the start point and end point of the simulation service. It does not participate in the establishment and maintenance of the PW. Therefore, it cannot perceive whether the end-to-end PW exists.

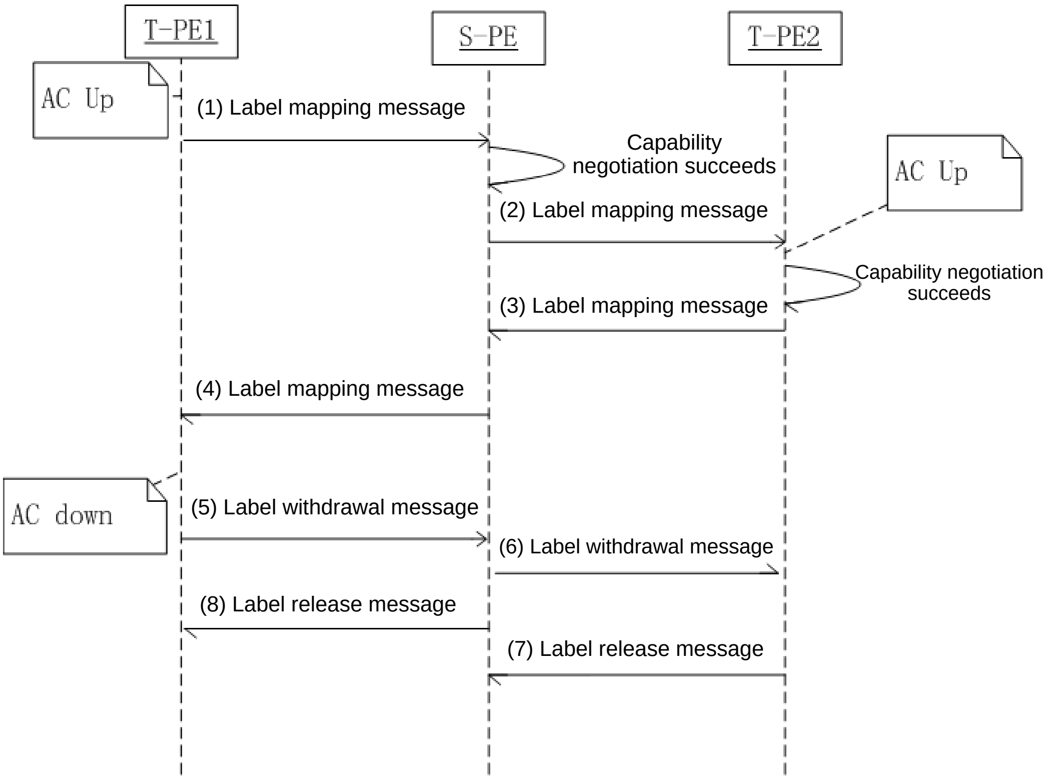
- T-PE (PW terminating provider edge): the device connected to the customer edge via the access circuit. This type of device mainly processes the data from the CE devices and forwards it to the peer PE devices. In addition, T-PE needs to process the simulation data from the peer end-to-end PW of the PW.

- S-PE (switching provider edge): a key device in the establishment of a MS-PW. It serves as the joint of the control plane and data plane of the MS-PW between two T-PEs and connects the adjacent segments of the MS-PW based on the PW identifier. On the data plane, the S-PE handles the multiplex layer of the PW and exchanges PW packets between adjacent PW segments. On the control plane, the S-PE participates in the establishment and maintenance of the PW. In addition, it participates in the VCCV and operation, administration, and management (OAM). You can also pre-configure the PW segments on the S-PE and bind the two PW segments on both sides of the S-PE.

Working Principle

The connection of multiple PW segments is not directly set up between two T-PE devices, but via an S-PE. For example, as shown in Figure 3-38, T-PE1 and T-PE2 set up a connection with the S-PE separately. The S-PE is responsible for connecting the PW segments on both sides of it. During establishment signaling negotiation, one of the T-PE devices sends a mapping packet that carries parameters to the S-PE, and the S-PE processes the packet and then forwards the processed packet to the other T-PE. After the parameters of both T-PE devices are negotiated, the PW is established. Same with the mapping packets, the release, withdrawal, and notification packets are also transmitted segment by segment.

Figure 54 MS-PW Signaling Interworking


T-PE1
S-PE
T-PE2
AC Up
(1) Label mapping message
Capability negotiation succeeds
AC Up
(2) Label mapping message
Capability negotiation succeeds
(3) Label mapping message
(4) Label mapping message
AC down
(5) Label withdrawal message
(6) Label withdrawal message
(8) Label release message
(7) Label release message

Implementation of the MS-PW solution is described in detail as follows:

- A PW segment connecting T-PE1 with the S-PE and a PW segment connecting T-PE2 with the S-PE are configured. During the signaling connection, T-PE1 and T-PE2 play the active role and the S-PE plays the passive role. However, if the S-PE is configured in static mode, it serves as the T-PE.

- The S-PE connects the two PW segments. During the signaling negotiation, the parameters of T-PE1 and T-PE2 are negotiated through the S-PE. If the parameters of both ends are consistent, the PW is established via the S-PE. Otherwise, the S-PE does not send label mapping messages to the downstream devices.

- Same with the mapping packets, the release, withdrawal, and notification packets are also transmitted segment by segment, so that the packets can be forwarded to each segment or one of the MS-PW segments can be removed without affecting others.

The figure above shows that T-PE1 initiates a delete or reset message. However, any PE device (T-PE or S-PE) on the MS-PW can possibly initiate the delete or reset message for certain reasons, causing the removal of the MS-PW. Similarly, the PE devices only need to send related messages to each PW segment of the MS-PW.

Protocols and Standards

- RFC5254: Requirements for Multi-Segment Pseudo-wire Emulation Edge-to-Edge (PWE3)

- RFC 6073: Segmented Pseudowire

- RFC 5659: An Architecture for Multi-Segment Pseudowire Emulation Edge-to-Edge

- RFC 5085: Pseudowire Virtual Circuit Connectivity Verification (VCCV): A Control Channel for Pseudowire

Defaults

The default PW value of the T-PE is the same as that of a common Martini VPWS and static VPWS. For the S-PE, when the PW is dynamically established, the related PW attributes of the default PW value are obtained via negotiation with the T-PE.

Configuring MS-PW

Perform the following steps to configure a MS-PW:

- (Mandatory) Configure the MPLS.
- (Mandatory) Configure PW segments.
- (Mandatory) Configure the PW switching point.
- Display the configuration.

Configuring MPLS

Enable the MPLS on the T-PEs and S-PE. The configuration is similar to the configuration of public network tunnels described in Section 3.2 "VPWS".

Configuring PW Segments

Configure the PW segments on both T-PE devices. The configuration is the same as that of the Martini VPWS and static VPWS. For details, see related sections.

Configuring PW Switching Point

Run the following commands to configure the PW switching point on S-PE:

Command	Function
Ruijie# **config terminal**	Enters the global configuration mode.
Ruijie(config)# **l2 vfi** *name* **switching**	Creates a switching VFI instance.
Ruijie(config-vfi)# **neighbor** *ip-address vc-id* **encapsulation mpls** [**manual**] [**no-switching-tlv**] [**tnl-policy** *policy-name*]	Joins the PWs located on both sides of the VFI.
Ruijie(config-vfi)# **end**	Returns to the privileged EXEC mode.
Ruijie# **copy running-config startup-config**	Saves the configuration.

Caution

The MS-PW can be established only after the PWs are configured successfully on both sides of S-PE.

Displaying Configuration

After the configuration is completed, run the following commands to display the information of the switching VFI:

Command	Function
Ruijie#**show mpls switching vfi** [*name*]	Displays the information of the switching VFI.

Ruijie#**show mpls l2transport switching vfi** [*name*]	Displays the switching information of the switching VFI.

Examples of Configuring MS-PW

Configuring Static MS-PW

Networking Requirements

You need to connect CE1 on Site A and CE2 on Site B to T-PE1 and T-PE2 respectively. The requirements are as follows:

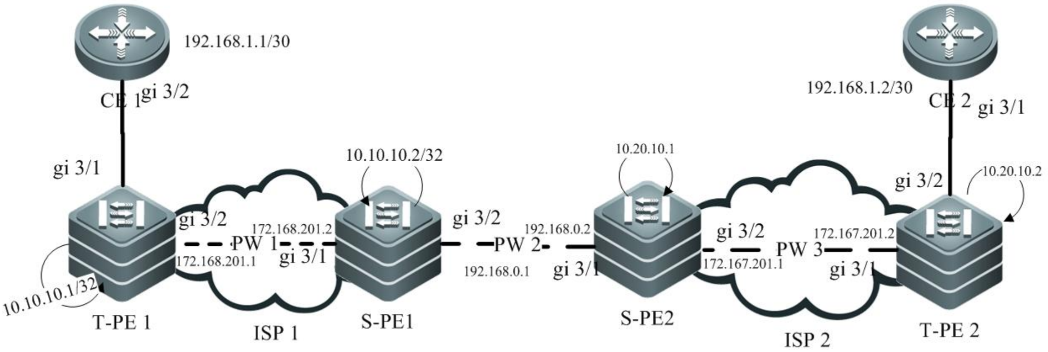
(1) T-PE1 and S-PE1 establish PW1.

(2) S-PE2 and T-PE2 establish PW3.

(3) Join PW1 and PW2 (from S-PE1 to S-PE2) on the switching node S-PE1 and join PW2 and PW3 (from S-PE1 to S-PE2) on the switching node S-PE2 to establish a cross-domain MS-PW.

Network Topology

Figure 55 Cross-domain MS-PW



Configuration Key Points

The configuration key points are as follows:

- The CE device is connected to the PE device through an Ethernet interface. The PW operates in Ethernet mode.
- The MPLS is configured on the backbone network to establish a label switched path (LSP) statically.
- PW segments are configured between T-PE1 and S-PE1 and between T-PE2 and S-PE2 separately.
- The PWs on both sides of the two switching node device are statically jointed to form a MS-PW.

Configuration Steps

- Configure CE-1.

#Configure the IP address.

```
Ruijie# config terminal
Ruijie(config)# interface gigabitethernet 3/2
Ruijie(config-if-GigabitEthernet 3/2)# ip address 192.168.1.1 255.255.255.252
Ruijie(config-if-GigabitEthernet 3/2)# exit
```

- Configure CE-2.
- The configuration of CE-2 is similar to that of CE-1.Configure T-PE1.

#Configure the IP address.

```
Ruijie# config terminal
Ruijie(config)# interface loopback 0
Ruijie(config-if-Loopback 0)# ip address 10.10.10.1 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
Ruijie(config)# interface gigabitEthernet 3/2
Ruijie(config-if-GigabitEthernet 3/2)# ip address 172.168.201.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 3/2) # exit
```

Ruijie(config)##Configure the routing protocol.

```
Ruijie(config)# router ospf 1
Ruijie(config-router)# network 10.10.10.1 0.0.0.0 area 0
Ruijie(config-router)# network 172.168.201.0 0.0.0.255 area 0
Ruijie(config-router)# exit
Ruijie(config)#
```

#Configure the MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-router)# ldp router-id interface loopback 0 force
Ruijie(config-router)# exit
Ruijie(config)# interface gigabitEthernet 3/2
Ruijie(config-if-GigabitEthernet 3/2)# label-switching
Ruijie(config-if-GigabitEthernet 3/2)# mpls ip
Ruijie(config-if-GigabitEthernet 3/2)# exit
Ruijie(config)#
```

#Bind the interface to the VPWS.

```
Ruijie(config)# interface gigabitEthernet 3/1
Ruijie(config-if-GigabitEthernet 3/1)# xconnect 10.10.10.2 10 encapsulation mpls
Ruijie(config-if-GigabitEthernet 3/1)# exit
Ruijie(config)#
```

#Configure the LDP extension peer.

```
Ruijie(config)# mpls router ldp
Ruijie(config-router)# neighbor 10.10.10.2
```

- Configure S-PE1.

#Configure the IP address.The configuration is similar to that of T-PE1:

#Configure the routing protocol.

```
Ruijie(config)# router ospf 1
Ruijie(config-router)# network 10.10.10.2 0.0.0.0 area 0
Ruijie(config-router)# network 172.168.201.0 0.0.0.255 area 0
Ruijie(config-router)# exit
```

#Configure the MPLS.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-router)# ldp router-id interface loopback 0 force
Ruijie(config-router)# exit
Ruijie(config)# interface gigabitEthernet 3/1
Ruijie(config-if-GigabitEthernet 3/1)# label-switching
Ruijie(config-if-GigabitEthernet 3/1)# mpls ip
Ruijie(config-if-GigabitEthernet 3/1)# exit
Ruijie(config)# interface gigabitEthernet 3/2
Ruijie(config-if-GigabitEthernet 3/2)# label-switching
Ruijie(config-if-GigabitEthernet 3/2)# mpls ip
Ruijie(config-if-GigabitEthernet 3/2)# exit
Ruijie(config)#
```

#Configure the switching VFI.

```
Ruijie(config)# l2 vfi ms-pw switching
Ruijie(config-vfi)# neighbor 10.10.10.1 10 encapsulation mpls
Ruijie(config-vfi)# neighbor 10.20.10.1 20 encapsulation mpls manual
Ruijie(config-vfi)# exit
Ruijie(config)#
```

#Configure the label information of the static PW.

```
Ruijie(config)# mpls static-l2vc 10.20.10.1 20 out-label 200 in-label 100
```

- Configure S-PE2.

The configuration is similar to that of S-PE1.

- Configure T-PE2.

The configuration is similar to that of T-PE1.

Displaying

Display the status of the MS-PW on S-PE1 and S-PE2. You can find that the MS-PW is in the up state.

```
Ruijie# show mpls switching vfi ms-pw
VFI name: ms-pw, Admin State: up
VFI Type: Switching, MTU: 1500
PW Encapsulation type: ethernet
Segment 1:
 Peer-Address  VC-ID    State  Local-label  Remote-label  Signal
 10.10.10.1      10      up      1500     1600         LDP
Segment 2:
 Peer-Address  VC-ID    State  Local-label  Remote-label  Signal
10.20.10.1     20       up       100        200       Static
```

Configuring MPLS GR

Note

Routers and router icons contained in this chapter refer to routers and layer-3 switches with the routing protocol enabled.

LDP GR

Overview

The Internet Engineering Task Force (IETF) has extended the Label Distribution Protocol (LDP), which is a signaling protocol of Multiprotocol Label Switching (MPLS), so that a device may instruct neighbors to keep related MPLS forwarding entries and set an "old" flag for these MPLS forwarding entries when the LDP is restarted on the device. After the LDP is restarted, neighbors assist the device to implement information synchronization so that the device is restored to the status before the LDP restart within the shortest possible time. The packet forwarding path does not change and data forwarding is not interrupted in the system throughout the LDP restart, guaranteeing the high reliability of MPLS application services.

Basic Concepts

GR Routers Classified by Capability

GR routers are classified by capability into GR-capable routers, GR-aware routers, and GR-unaware routers.

- GR-Capable Router

A GR-capable router is a router that has GR capability. In general, a GR-capable router is equipped with two management boards which work in 1+1 master/slave mode. The GR-capable router can send an advertisement packet to neighboring routers during master/slave switchover of the management boards, so that neighboring routers keep forwarding entries related to the GR-capable router. After master/slave switchover, routing tables are re-established without causing route flapping or changing the packet forwarding path, guaranteeing uninterrupted data forwarding in the system.

- GR-Aware Router

A GR-aware router is a router that has GR detection capability. It may be not equipped with two management boards but can detect that its neighbors are experiencing GR and can assist its neighbors to complete GR.

- GR-Unaware Router

A GR-unaware router is a router that does not have GR detection capability. It cannot detect that its neighbors are experiencing GR, cannot assist its neighbors to complete GR, and does not have GR capability. Generally, if a router has no awareness capability, it is because the system software does not provide the GR feature or the GR feature is disabled.

GR Routers Classified by Role

GR routers are classified by role during router restart into GR restarters and GR helpers.

- GR Restarter

The GR restarter has GR capability and its restart is triggered by administrators or faults.

- GR Helper

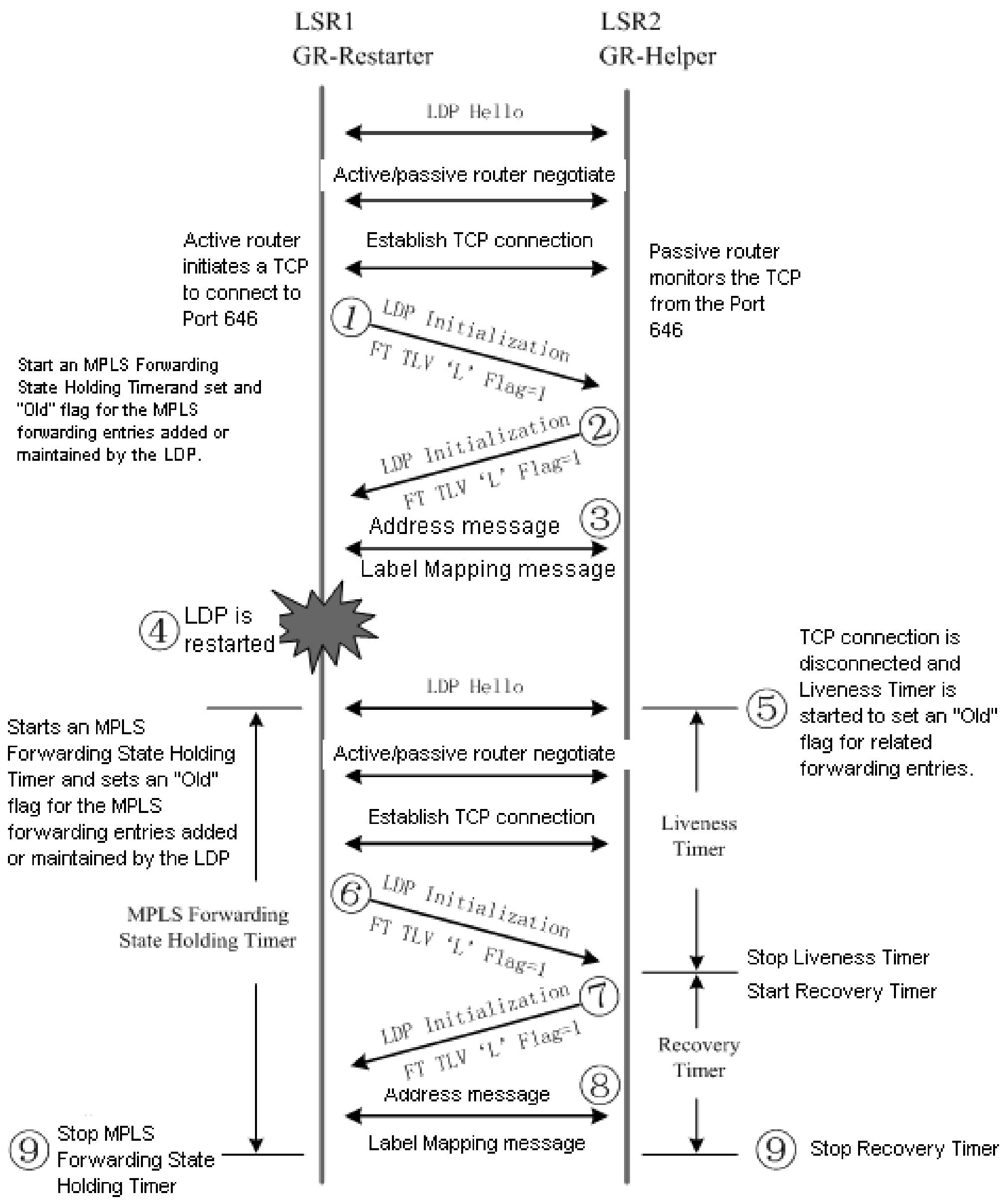
The GR helper is a neighbor of the GR restarter. It must be at least a GR-aware router.

Working Principle

The LDP GR function must be enabled on and supported by two routers in order to establish a GR-capable LDP session between them. If not, only a common LDP session can be established. If the initiator supports LDP GR and LDP GR is enabled on it during LDP session establishment, the initiator sends an Initialization message that carries an FT Session TLV.

If the passive router receives an Initialization message that carries an FT Session TLV during session establishment, it may choose to add or not add the FT Session TLV to the Initialization message to be sent to the initiator depending on its LDP GR capability. If the passive router supports LDP GR and LDP GR is enabled on it, the Initialization message will carry the FT Session TLV to establish a GR-capable LDP session. Otherwise, the Initialization message will not carry the FT Session TLV to establish a common GR-incapable LDP session LDP session. If the passive router receives an Initialization message that does not carry the FT Session TLV, a common GR-incapable LDP session is established, no matter whether the passive router adds the FT Session TLV to the Initialization message to be sent. Figure 1-1 shows the process of LDP session establishment between two Label Switching Routers (LSRs) with LDP GR capability.

Figure 1-1 LDP GR session establishment process


LSR1
GR-Restarter
LSR2
GR-Helper
LDP Hello
Active/passive router negotiate
Establish TCP connection
Active router initiates a TCP to connect to Port 646
Passive router monitors the TCP from the Port 646
1
LDP Initialization
FT TLV 'L' Flag=1
Start an MPLS Forwarding State Holding Timerand set and "Old" flag for the MPLS forwarding entries added or maintained by the LDP.
2
LDP Initialization
FT TLV 'L' Flag=1
3
Address message
Label Mapping message
4
LDP is restarted
LDP Hello
TCP connection is disconnected and Liveness Timer is started to set an "Old" flag for related forwarding entries.
5
Starts an MPLS Forwarding State Holding Timer and sets an "Old" flag for the MPLS forwarding entries added or maintained by the LDP
Active/passive router negotiate
Establish TCP connection
Liveness Timer
MPLS Forwarding State Holding Timer
6
LDP Initialization
FT TLV 'L' Flag=1
Stop Liveness Timer
Start Recovery Timer
7
LDP Initialization
FT TLV 'L' Flag=1
Recovery Timer
8
Address message
Label Mapping message
9
Stop MPLS Forwarding State Holding Timer
9
Stop Recovery Timer

LSR 1 initiates an Initialization message that carries an optional FT Session TLV, indicating that LSR 1 itself supports LDP GR.

After LSR 2 that supports LDP GR receives the Initialization message that carries the FT Session TLV, it returns an Initialization message that also carries the FT Session TLV to LSR 1. When LSR 1 receives the Initialization message from LSR 2, a GR-capable LDP session is established.

LSR 1 and LSR 2 exchange LDP Address messages and Label Mapping messages with each other.

The LDP on LSR 1 is restarted for a certain reason. LSR 1 keeps all MPLS forwarding entries added or maintained by the LDP, sets an "old" flag for these forwarding entries, and starts an MPLS Forwarding State Holding Timer.

After LSR 2 with LDP GR capability detects that the GR-capable LDP session with LSR 1 is disconnected, LSR 2 keeps the MPLS forwarding entries related to this session and sets an "old" flag for these forwarding entries. At the same time, LSR 2 selects a smaller value of its Liveness Timer and the FT Reconnect Timeout in the received FT Session TLV to start a Liveness Timer, and keeps these "old" forwarding entries before the Liveness Timer is triggered.

To re-establish a session with LSR 2, LSR 1 sets the Recovery Time in the FT Session TLV of the Initialization message to the remaining value of the MPLS Forwarding State Holding Timer.

After receiving the Initialization message that carries the FT Session TLV from LSR 1, LSR 2 detects that Recovery Time is not 0, LSR 2 continues to keep the "old" forwarding entries, stops the Liveness Timer, selects a smaller value of the its Recovery Time and the Recovery Time in the received FT Session TLV to start a Recovery Timer, and keeps these "old" forwarding entries before the Recovery Timer is triggered.

LSR 1 and LSR 2 exchange LDP Address and Label Mapping messages with each other, and keep or remove the "old" flag set for MPLS forwarding entries.

The GR process ends. LSR 1 and LSR 2 delete the MPLS forwarding entries with the "old" flag from themselves respectively.

Protocols and Specifications

The protocols or specifications involved are as follows:

- RFC 3036: LDP Specification
- RFC 3037: LDP Applicability
- RFC 3215: LDP State Machine
- RFC 3478: Graceful Restart Mechanism for LDP
- RFC 3479: Fault Tolerance for LDP

Configuring LDP GR

Network Environment

The GR function of the MPLS LDP is configured to maintain the neighborhood and sessions between routers and recover sessions and label information when faults occur on MPLS devices.

Prerequisites

Complete the following tasks before configuring MPLS LDP GR:

- Configure IGP GR.
- Configure MPLS LDP session information.

Data Preparations

Prepare the following data before configuring MPLS LDP GR:

- LDP session re-connection time
- LDP neighbor keep-alive time
- LDP session recovery time

Configuring the LDP GR

By default, the LDP GR function is disabled on a device. To enable the LDP GR function on a device, enter privileged user mode and run the following commands:

Command	Function
Ruijie#**configure terminal**	Enters global configuration mode.

Ruijie(config)#**mpls ip**	Enables global MPLS forwarding. Caution This command is not applicable to forwarding on switch chips.
Ruijie(config)#**interface type ID**	Enters interface configuration mode.
Ruijie(config-if-type ID)#**no switchport**	Sets ports to L3 ports (Switch configuration).
Ruijie(config-if-type ID)#**mpls ip**	Enables LDP forwarding on interfaces.
Or:	
Ruijie(config-if-type ID)#**mpls ip**	Enables LDP forwarding on interfaces (Router configuration).
Ruijie(config-if-type ID)#**ip ref**	Enables fast forwarding on interfaces.
Ruijie(config-if-type ID)#**label-switching**	Enables MPLS packet processing on interfaces.
Ruijie(config)#**mpls router ldp**	Enters LDP configuration mode.
Ruijie(config-mpls-router)#l**dp router-id** *Loopback ID* **force**	Sets the router ID to a loopback ID (The settings immediately take effect).
Ruijie(config-mpls-router)#**graceful-restart**	Enables LDP GR. By default, LDP GR is disabled.
Ruijie(config-mpls-router)#**end**	Exits LDP configuration mode.
Ruijie#s**how mpls ldp graceful-restart**	Displays LDP GR sessions and session parameters.

To disable LDP GR, run the **no graceful-restart** command.

Note The existing LDP session is not affected when LDP GR is enabled. For example, the LDP session does not restart. The LDP GR takes effect only after the LDP session restarts..

Configuring Parameters Related to LDP GR (Optional)

Enter LDP configuration mode and run the following commands:

Command	**Function**
Ruijie#**configure terminal**	Enters global configuration mode.
Ruijie(config)#**mpls router ldp**	Enables the LDP and enters LDP configuration mode.
Ruijie(config-mpls-router)**#graceful-restart timer reconnect** *seconds*	Sets the LDP session re-connection time. By default, the LDP session reconnection time is 300 seconds.
Ruijie(config-mpls-router)#**graceful-restart timer neighbor-liveness** *seconds*	Sets the LDP neighbor keep-alive time. By default, the LDP neighbor keep-alive time is 120 seconds.
Ruijie(config-mpls-router)#**graceful-restart timer recovery** *seconds*	Sets the LDP session recovery time. By default, the LDP session recovery time is 120 seconds.
Ruijie(config-mpls-router)#**end**	Exits LDP configuration mode.
Ruijie#**show mpls ldp graceful-restart**	Displays LDP GR sessions and session parameters.

To restore the default settings of parameters related to LDP GR, run the **no graceful-restart timer reconnect, no graceful-restart timer neighbor-liveness**, and **no graceful-restart timer recovery** commands.

Verification

To display LDP GR configuration and running information, run the following commands.

Command	Function
show mpls ldp graceful-restart	Displays LDP GR sessions and session parameters.
show mpls ldp bindings [all | **vrf** *vrf-name*] [*ip-address/mask*|label *label*] **[remote | local**]	Displays the mapping between Forwarding Equivalence Classes (FECs) and labels.
show mpls ldp neighbor [all | **vrf** *vrf-name*] [*ip-address*] [**detail]**	Displays the status of LDP neighbors.

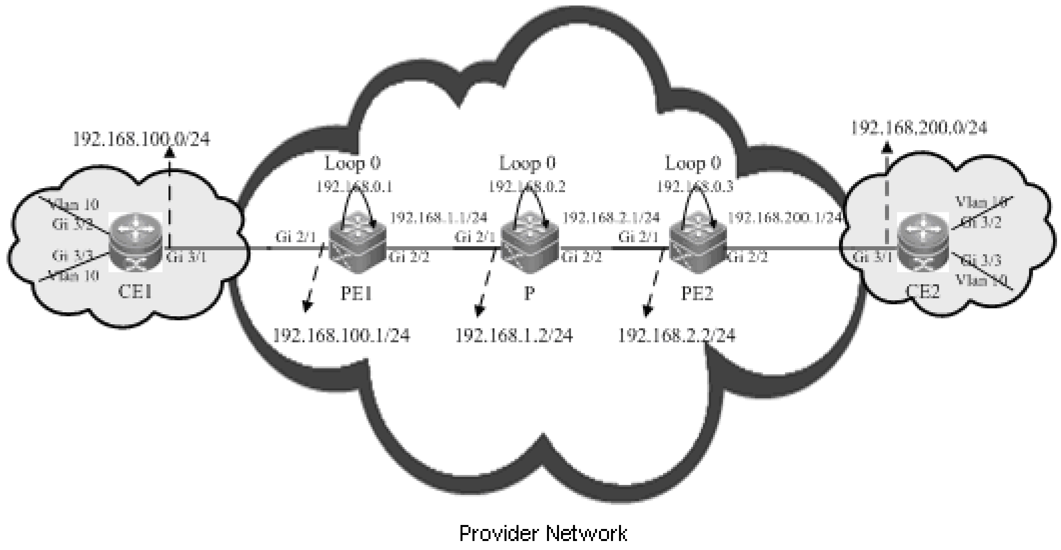
Configuration Examples

Networking Requirements

- An MPLS network consists of Provider Edge (PE) and Provider (P) devices.
- PE and P devices support the LDP and are capable of GR.
- This following describes how to configure LDP GR on PE 1 and a P device. PE 1 is a GR-capable router and the GR restarter. The P device is a GR-aware router and the GR helper.

Networking Topology

Figure 1-2 Networking topology for configuring LDP GR



Configuration Tips

Configure PE 1 and the P device as follows:

- Configure interface IP addresses and the Open Shortest Path First (OSPF) protocol.
- Enable global MPLS packet forwarding on devices, MPLS forwarding, and the LDP on interfaces.
- Configure the LDP to enable the network to forward MPLS traffic.
- Enable the LDP GR protocol.

- Configure parameters related to LDP GR.
- Restart the LDP session for the configurations to take effect.

Configuration Steps

- Configure interface IP addresses and the OSPF protocol.

# Configure PE 1.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
```

## The **no switchport** command is used on a switch to switch to Routed Port mode. It is not applicable to routers, and therefore you do not need to run this command on routers.

```
Ruijie(config)#interface gigabitEthernet 2/1
Ruijie(config-if-GigabitEthernet 2/1)#no switchport
Ruijie(config-if-GigabitEthernet 2/1)#ip address 192.168.100.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 2/1)#exit
```

## The **no switchport** command runs on a switch to switch to Routed Port mode. It is not applicable to routers, and therefore you do not need to run this command on routers.

```
Ruijie(config)#interface gigabitEthernet 2/2
Ruijie(config-if-GigabitEthernet 2/2)#no switchport
Ruijie(config-if-GigabitEthernet 2/2)#ip address 192.168.1.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 2/2)#exit
Ruijie(config)#interface loopback 0
Ruijie(config-Loopback 0)#ip address 192.168.0.1 255.255.255.255
Ruijie(config-Loopback 0)#exit
Ruijie(config)#router ospf 1
Router(config-router)#network 192.168.100.1 255.255.255.0 area 0
Router(config-router)#network 192.168.1.1 255.255.255.0 area 0
Router(config-router)#network 192.168.0.1 255.255.255.255 area 0
Router(config-router)#exit
```

# Configure the P device by running the same commands as those on PE 1.

- Enable global MPLS forwarding on devices, MPLS packet forwarding, and the LDP on interfaces.

# Configure PE 1.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#mpls ip
```

## The **ip ref** command is used on a router to enable MPLS fast forwarding on the router. You do not need to run this command on switches.

```
Ruijie(config)#interface gigabitEthernet 2/2
Ruijie(config-if-GigabitEthernet 2/2)#label-switching
Ruijie(config-if-GigabitEthernet 2/2)#mpls ip
Ruijie(config-if-GigabitEthernet 2/2)#ip ref
```

```
Router(config-if-GigabitEthernet 2/2)#exit
```

# Configure the P device.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#mpls ip
```

## The **ip ref** command is used on a router to enable MPLS fast forwarding on the router. You do not need to run this command on switches.

```
Ruijie(config)#interface gigabitEthernet 2/1
Ruijie(config-if-GigabitEthernet 2/1)#label-switching
Ruijie(config-if-GigabitEthernet 2/1)#mpls ip
Ruijie(config-if-GigabitEthernet 2/1)#ip ref
Router(config-if-GigabitEthernet 2/1)#exit
```

## The **ip ref** command is used on a router to enable MPLS fast forwarding on the router. You do not need to run this command on switches.

```
Ruijie(config)#interface gigabitEthernet 2/2
Ruijie(config-if-GigabitEthernet 2/2)#label-switching
Ruijie(config-if-GigabitEthernet 2/2)#mpls ip
Ruijie(config-if-GigabitEthernet 2/2)#ip ref
Router(config-if-GigabitEthernet 2/2)#exit
```

- Configure the LDP to enable the network to forward MPLS traffic.

# Configure PE 1.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#mpls router ldp
Ruijie(config-mpls-router)#ldp router-id interface loopback 0 force
```

# Configure the P device.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#mpls router ldp
Ruijie(config-mpls-router)#ldp router-id interface loopback 0 force
```

- Enable the LDP GR protocol.

# Configure PE 1.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#mpls router ldp
Ruijie(config-mpls-router)#graceful-restart
```

# Configure the P device.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
```

```
Ruijie(config)#mpls router ldp
Ruijie(config-mpls-router)#graceful-restart
```

- Configure parameters related to LDP GR.

# Configure PE 1.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#mpls router ldp
```

## Set the LDP reconnection time to 300 seconds, LDP neighbor keep-alive time to 120 seconds, and LDP recovery time to 120 seconds.

```
Ruijie(config-mpls-router)#graceful-restart timer reconnect 300
Ruijie(config-mpls-router)#graceful-restart timer neighbor-liveness 120
Ruijie(config-mpls-router)#graceful-restart timer recovery 120
Ruijie(config-mpls-router)#exit
```

# Configure the P device.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#mpls router ldp
```

## Set the LDP reconnection time to 300 seconds, LDP neighbor keep-alive time to 120 seconds, and LDP recovery time to 120 seconds.

```
Ruijie(config-mpls-router)#graceful-restart timer reconnect 300
Ruijie(config-mpls-router)#graceful-restart timer neighbor-liveness 120
Ruijie(config-mpls-router)#graceful-restart timer recovery 120
Ruijie(config-mpls-router)#exit
```

- Restart the LDP session for the configurations to take effect.

# Restart the LDP session on PE 1.

```
Ruijie#clear mpls ldp neighbor all
```

# Restart the LDP session on the PE device.

```
Ruijie#clear mpls ldp neighbor all
```

Verification

Run the following commands to view configurations on PE 1:

# View LDP GR information on PE 1.

```
Ruijie#show mpls ldp graceful-restart
Default VRF:
  LDP Graceful Restart is enabled
  Neighbor Liveness Timer: 120 seconds
  Max Recovery Time: 120 seconds
  Forwarding State Holding Time: 300 seconds
  Down Neighbor Database (1 records):
   Peer LDP Ident: 192.168.0.2:0; Local LDP Ident: 192.168.0.1:0
       Status: recovering (86 seconds left)
       Address list contains 3 addresses:
        192.168.1.2   192.168.2.1   192.168.0.2
  Graceful Restart-enabled Sessions:
   Peer LDP Ident: 192.168.0.2:0, State: estab
```

# View LDP GR neighbor information on PE 1.

```
Ruijie#show mpls ldp neighbor
Default VRF:
   Peer LDP Ident: 192.168.0.2:0; Local LDP Ident: 192.168.0.1:0
       TCP connection: 192.168.0.2.15532 - 192.168.0.1.646
       State: OPERATIONAL; Msgs sent/recv: 23/27; UNSOLICITED
       Up time: 00:04:12
       Graceful Restart enabled; Peer reconnect time (msecs): 0
```

# View LDP binding information on PE 1.

```
Router#show mpls ldp bindings
Default VRF:
   lib entry: 192.168.0.2/32
       local binding:  to lsr:   192.168.0.2:0, label: 1024
       remote binding: from lsr: 192.168.0.2:0, label: imp-null stale
lib entry: 192.168.1.2/24
       local binding:  to lsr:   192.168.0.2:0, label: 1025
       remote binding: from lsr: 192.168.0.2:0, label: imp-null stale
   lib entry: 192.168.2.1/24
       local binding:  to lsr:   192.168.0.2:0, label: 1026
       remote binding: from lsr: 192.168.0.2:0, label: imp-null stale
```

L3VPN GR

Overview

L3 VPN GR (VPN GR) implements uninterrupted forwarding of Virtual Private Network (VPN) services. It ensures that the data of VPN services can be normally forwarded when the control plane on a device fails, protecting VPN services on the network.

The following prerequisites must be met for VPN GR:

- Devices support 1+1 management board redundancy.
- Devices support uninterrupted forwarding of routing protocols.
- Devices support the BGP/MPLS GR protocol.
- Devices support the LDP GR protocol.

The objectives of VPN GR are to:

- Minimize routing protocol flapping during master/slave management board switchover.

- Minimize the impact on VPN services.
- Minimize Single-Point Failures (SPFs) on access devices and improve VPN network reliability.
- Minimize the packet loss rate of VPN traffic.

Note

GR must be supported in unicast routing to implement uninterrupted forwarding of routing protocols. In other words, the device must support OSPF GR, IS-IS GR, or BGP GR.

Basic Concepts

GR Routers Classified by Capability

GR routers are classified by capability into GR-capable routers, GR-aware routers, and GR-unaware routers.

- GR-Capable Router

A GR-capable router is a router that has GR capability. In general, a GR-capable router is equipped with two management boards which work in 1+1 master/slave mode. The GR-capable router can send an advertisement packet to neighboring routers during master/slave switchover of the management boards, so that neighboring routers keep forwarding entries related to the GR-capable router. After master/slave switchover, routing tables are re-established without causing route flapping or changing the packet forwarding path, guaranteeing uninterrupted data forwarding in the system.

- GR-Aware Router

A GR-aware router is a router that has GR detection capability. It may be not equipped with two management boards but can detect that its neighbors are experiencing GR and can assist its neighbors to complete GR.

- GR-Unaware Router

A GR-unaware router is a router that does not have GR detection capability. It cannot detect that its neighbors are experiencing GR, cannot assist its neighbors to complete GR, and does not have GR capability. Generally, if a router has no awareness capability, it is because the system software does not provide the GR feature or the GR feature is disabled.

GR Routers Classified by Role

GR routers are classified by role during router restart into GR restarters and GR helpers.

- GR Restarter

The GR restarter has GR capability and its restart is triggered by administrators or faults.

- GR Helper

The GR helper is a neighbor of the GR restarter. It must be at least a GR-aware router.

Working Principle

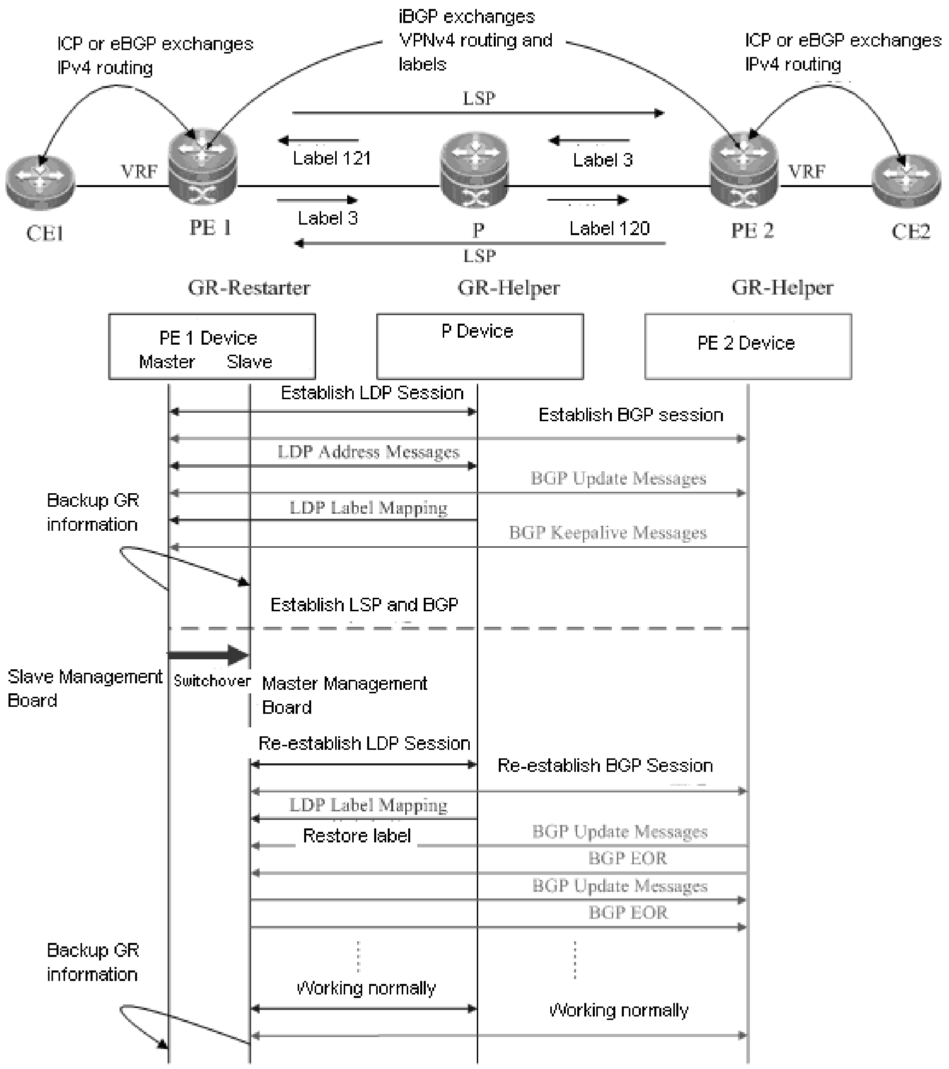
Both the control plane and the forwarding plane on a traditional device are implemented by the same processor which simultaneously maintains a routing table and a forwarding table. To improve forwarding performance and reliability of devices, a multi-processor architecture is used on high-end and mid-range devices. The processors of control modules such as the routing protocol module are located on a master management board and data forwarding processors are located on line cards. In this manner,, the control plane and the forwarding plane are separated from each other so that data forwarding on line cards is not affected when the control plane is restarted. This technology provides a prerequisite for implementing GR. The GR-capable routers mentioned in this document are such routers where the control plane and the forwarding plane are separated from each other.

The VPN network shown in Figure 1-3 has the following features:

- Customer Edge (CE) devices represent a customer network. The Interior Gateway Protocol (IGP) or Exterior Border Gateway Protocol (eBGP) runs on the CE devices.
- PE and P devices form a provider network. The IGP runs on these devices.
- Public network tunnels and Label Switching Paths (LSPs) are established using the LDP between PE 1, PE 2, and the P device.
- Private network tunnels are established using the Interior Border Gateway Protocol (iBGP) between PE 1 and PE 2.
- The IGP, BGP, and LDP have GR capability.
- PE devices are GR-capable routers, and the P device is a GR-aware router.

Note For details about IGP GR, BGP GR, and LDP GR, see related sections in *OSPF*, *BGP*, and *LDP GR*.

Figure 1-3 VPN networking topology



The following describes the GR process of the access device PE 1 of a provider network. Master/slave management board switchover occurs on PE 1. PE 1 works as a GR restarter and PE 2 and the P device works as GR helpers.. The process of master/slave management board switchover is as follows:

- Before master/slave management board switchover

PE 1 performs IGP GR or eBGP GR negotiation with the connected CE device, IGP GR and LDP GR negotiation with the P device, and iBGP GR negotiation with PE 2.

PE 1 sends an Initialization message that carries the optional FT Session TLV to the P device to establish a GR-capable LDP session. After the LDP session is established, they exchange LDP Address messages and Label Mapping messages with each other. In this manner, GR-capable LSPs are established for data forwarding.

PE 1 sends an Open message to PE 2 to establish a GR-capable iBGP session. The Open message carries GR capability parameters <AFI=IPv4, SAFI=Unicast> and <AFI=IPv4, SAFI=VPNv4>.

When the master management board works, the GR information is backed up to the slave management board. In this manner, the GR backup enables the system to access these original data during GR protocol restart and apply the data to the protocol GR process after master/slave management board switchover.

- During master/slave management board switchover

After the GR information on PE 1 is backed up to the slave management board,, the master/slave management board switchover is performed.

After detecting that the Transfer Control Protocol (TCP) session is down, the P device sets an "old" flag for the respective LSPs and starts a forwarding entry aging timer to continue to forward data before this aging timer expires.

After detecting that the TCP connection is broken, PE 2 immediately marks the route learned from PE 1 as "old" and starts a restart timer for PE 1. If PE 2 does not receive an Open message within the restart timer, it deletes the "old" flag for the route. If PE 2 receives an Open message, it deletes the restart timer. In this period of time, PE 1 and PE 2 continue to forward traffic along the original route.

- After master/slave management board switchover

The slave management board on PE 1 becomes the new master management board, and the original master management board becomes the new slave management board. The new master management board checks the GR information backed up and determines whether forwarding entries before the restart are retained. Then the Command Line Interface (CLI) configuration initialization process and the GR process continue. During IGP GR, BGP GR, and LDP GR, all the devices send notification messages to all IGP, BGP, and LDP neighbors for connection re-establishment.

113) IGP convergence

PE 1 sends an Initialization message that carries the FT Session TLV to the P device, and re-establishes a session upon receiving a response to obtain topology and routing information. PE 1 re-computes the routing table and deletes the "old" routes. The IGP convergence is completed.

114) BGP processing

PE 1 and CE 1 exchange routing information with each other. PE 1 updates its routing table and forwarding entries based on the new routing and forwarding information, substitutes invalid routes. The BGP convergence is completed.

PE 1 and PE 2 start to re-establish a BGP session with each other. PE 1 sends an Open message to PE 2. The Open message carries GR capability parameters. PE 1 receives and processes an Update message from PE 2. These messages carry IP prefix information. PE 1 does not start BGP route preference until receiving an EOR flag from PE2. PE 1 sends an Update message that carries prefix information to PE 2. After sending the Update message, PE 1 sends the EOR flag to PE 2. After receiving the EOR flag, PE 2 starts BGP route preference. The network convergence is completed.

115) LDP processing

PE 1 sends an Initialization message that carries the FT Session TLV to the neighboring P device. A GR-capable LDP session is established after the P device receives the Initialization message. Then PE 1 and the P device exchange LDP Address and Label Mapping messages again with each other, and keep or remove the "old" flag set for MPLS forwarding entries. When the GR process ends, both devices delete their "old" MPLS forwarding entries respectively.

Note The preceding IGP GR, BGP GR, and LDP GR processes do not follow a strict priority sequence. In terms of route convergence, unicast routes converge first and converged routes are advertised to the LDP.

Before all protocols complete the GR process, only Routing Information Base (RIB) information on the master management board is updated. The Forwarding Information Base (FIB) information on interface boards is not updated.

The FIB information on interface boards is updated only after all protocols complete the GR process.

Protocols and Specifications

The protocols or specifications involved are as follows:

- RFC 4724: Graceful Restart Mechanism for BGP
- RFC 4781: Graceful Restart Mechanism for BGP with MPLS

Configuring L3VPN GR

Network Environment

On an MPLS network, L3VPN GR is required for the devices carrying L3VPN services. In this manner, data forwarding is not interrupted during master/slave management board switchover on devices, guaranteeing traffic continuity.

Note The GR capability does not guarantee traffic continuity when master/slave management board switchover also occurs on neighboring devices.

Prerequisites

Complete the following tasks before configuring L3VPN GR:

- Build an L3VPN environment and configure L3VPN.
- Ensure that devices support management board redundancy.
- Ensure that the IGP has GR capability.
- Ensure that the BGP has GR capability.
- Ensure that the LDP has GR capability.

Note For details about L3VPN configuration, see related sections in *BGP/MPLS VPN Configuration*.

Data Preparations

Prepare the following data before configuring L3VPN GR:

- IGP GR parameters
- BGP GR parameters
- LDP GR parameters

Configuring IGP GR

Note For details about IGP GR configuration, see related sections in *OSPF*.

Configuring BGP GR

Note For details about BGP GR configuration, see related sections in *BGP*.

Configuring LDP GR

Note For details about LDP GR configuration, see the "LDP GR" section in this document.

Verification

To display L3VPN GR configuration and running information, run the following commands:

Command	Function
show ip vrf [*vrf_name*]	Displays VRF configuration information.
show ip bgp vpnv4 {**all** | **rd** *route-distinguish* | **vrf** *vrf_name*} [*network-address*] **[summary**] [**neighbor] [label]**	Displays VPN routing information.
show ip bgp summary	Displays the status of all BGP connections.
show ip route vrf *vrf_name* [**A.B.C.D** | **bgp** | c**onnected | isis | ospf | rip | static | weight]**	Displays VRF routing information.
show mpls ldp graceful-restart [all | vrf *vrf-name***]**	Displays LDP GR sessions and session parameters.

Note All the preceding commands can be configured in any mode except for user mode.

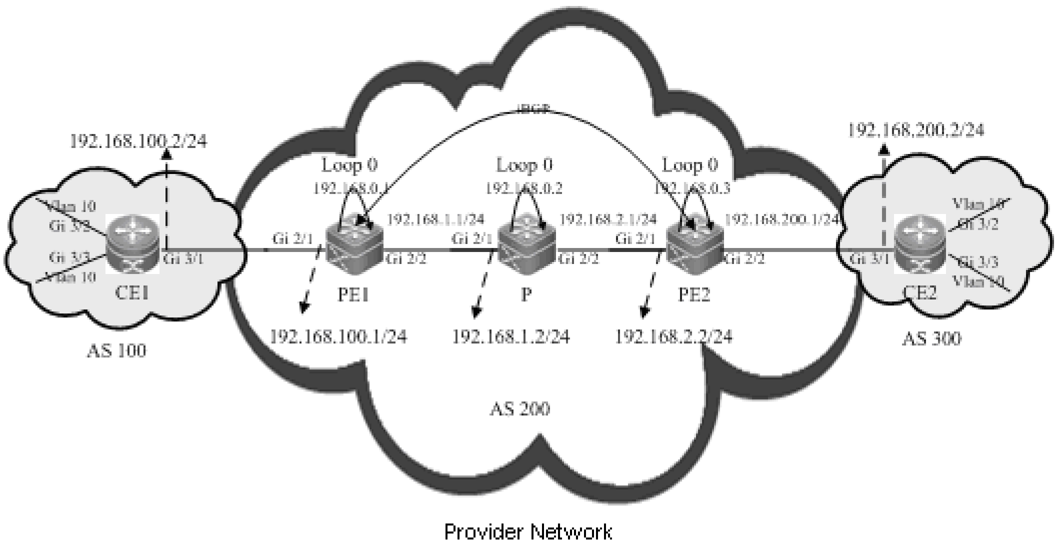
Configuration Examples

Networking Requirements

- CE devices represent a customer network. The IGP or eBGP runs on the CE devices.
- PE and P devices form a provider network. The IGP runs on these devices.
- Public network tunnels and LSPs are established using the LDP between PE 1, PE 2, and the P device.
- Private network tunnels are established using the iBGP between PE 1 and PE 2.
- The IGP, BGP, and LDP have GR capability.
- PE devices are GR-capable routers, and the P device is a GR-aware router.

Networking Topology

Figure 1-4 Networking topology for configuring L3VPN GR



Configuration Tips

Configure PE 1, PE 2, and the P device as follows:

- Configure VRF.
- Configure interface IP addresses and the OSPF protocol.
- Enable global MPLS forwarding on devices, MPLS packet forwarding, and the LDP on interfaces.
- Configure the LDP to enable the network to forward MPLS traffic.
- Enable the LDP GR protocol and configure parameters related to LDP GR.
- Configure L3VPN.
- Enable the BGP GR protocol.
- Restart the LDP session for the configurations to take effect.

Configuration Steps

- Configure VRF.

# Configure PE 1.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
```

## Define VRF.

```
Ruijie(config)#ip vrf 10
Ruijie(config-vrf)#rd 1:100
Ruijie(config-vrf)#route-target both 1:100
Ruijie(config-vrf)#exit
```

# Configure the P device. VRF configuration is not required on the P device.

# Configure PE 2 by running the same commands as those on PE 1.

- Configure interface IP addresses and the OSPF protocol.

# Configure PE 1.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
```

## The **no switchport** command is used on a switch to switch to Routed Port mode. It is not applicable to routers and therefore you do not need to run the command on routers.

```
Ruijie(config)#interface gigabitEthernet 2/1
Ruijie(config-if-GigabitEthernet 2/1)#no switchport
Ruijie(config-if-GigabitEthernet 2/1)#ip vrf forwarding 10
Ruijie(config-if-GigabitEthernet 2/1)#ip address 192.168.100.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 2/1)#exit
```

## The **no switchport** command is used on a switch to switch to Routed Port mode. It is not applicable to routers and therefore you do not need to run the command on routers.

```
Ruijie(config)#interface gigabitEthernet 2/2
Ruijie(config-if-GigabitEthernet 2/2)#no switchport
Ruijie(config-if-GigabitEthernet 2/2)#ip address 192.168.1.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 2/2)#exit
```

## Configure the loopback interface **loopback 0**.

```
Ruijie(config)#interface loopback 0
Ruijie(config-Loopback 0)#ip address 192.168.0.1 255.255.255.255
Ruijie(config-Loopback 0)#exit
```

## Activate the OSPF protocol and enter OSPF mode.

```
Ruijie(config)#router ospf 10
Ruijie(config-router)#network 192.168.0.1 255.255.255.255 area 0
Ruijie(config-router)#network 192.168.1.0 255.255.255.0 area 0
Ruijie(config-router)#end
```

# Configure the P device.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
```

## The **no switchport** command is used on a switch to switch to Routed Port mode. It is not applicable to routers and therefore you do not need to run the command on routers.

```
Ruijie(config)#interface gigabitEthernet 2/1
Ruijie(config-if-GigabitEthernet 2/1)#no switchport
Ruijie(config-if-GigabitEthernet 2/1)#ip address 192.168.1.2 255.255.255.0
Ruijie(config-if-GigabitEthernet 2/1)#exit
```

## The **no switchport** command is used on a switch to switch to Routed Port mode. It is not applicable to routers and therefore you do not need to run the command on routers.

```
Ruijie(config)#interface gigabitEthernet 2/2
Ruijie(config-if-GigabitEthernet 2/2)#no switchport
Ruijie(config-if-GigabitEthernet 2/2)#ip address 192.168.2.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 2/2)#exit
```

## Configure the loopback interface **loopback 0**.

```
Ruijie(config)#interface loopback 0
Ruijie(config-Loopback 0)#ip address 192.168.0.2 255.255.255.255
Ruijie(config-Loopback 0)#exit
```

## Activate the OSPF protocol and enter OSPF mode.

```
Ruijie(config)#router ospf 10
Ruijie(config-router)#network 192.168.1.0 255.255.255.0 area 0
Ruijie(config-router)#network 192.168.2.0 255.255.255.0 area 0
Ruijie(config-router)#network 192.168.0.2 255.255.255.255 area 0
Ruijie(config-router)#end
```

# Configure PE 2 by running the same commands as those on PE 1.

- Enable global MPLS forwarding on devices, MPLS packet forwarding, and the LDP on interfaces.

# Configure PE 1.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#mpls ip
```

## The **ip ref** command is used on a router to enable MPLS fast forwarding on the router. You do not need to run this command on switches.

```
Ruijie(config)#interface gigabitEthernet 2/2
Ruijie(config-if-GigabitEthernet 2/2)#label-switching
Ruijie(config-if-GigabitEthernet 2/2)#mpls ip
Ruijie(config-if-GigabitEthernet 2/2)#ip ref
Router(config-if-GigabitEthernet 2/2)#exit
```

# Configure the P device.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
```

```
Ruijie(config)#mpls ip
```

## The **ip ref** command is used on routers to enable MPLS fast forwarding on the router. You do not need to run this command on switches.

```
Ruijie(config)#interface gigabitEthernet 2/1
Ruijie(config-if-GigabitEthernet 2/1)#label-switching
Ruijie(config-if-GigabitEthernet 2/1)#mpls ip
Ruijie(config-if-GigabitEthernet 2/1)#ip ref
Router(config-if-GigabitEthernet 2/1)#exit
```

## The **ip ref** command is used on a router to enable MPLS fast forwarding on the router. You do not run this command on switches.

```
Ruijie(config)#interface gigabitEthernet 2/2
Ruijie(config-if-GigabitEthernet 2/2)#label-switching
Ruijie(config-if-GigabitEthernet 2/2)#mpls ip
Ruijie(config-if-GigabitEthernet 2/2)#ip ref
Router(config-if-GigabitEthernet 2/2)#exit
```

# Configure PE 2 by running the same commands as those on PE 1.

- Configure the LDP to enable the network to forward MPLS traffic.

# Configure PE 1.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#mpls router ldp
Ruijie(config-mpls-router)#ldp router-id interface loopback 0 force
```

# Configure the P device.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#mpls router ldp
Ruijie(config-mpls-router)#ldp router-id interface loopback 0 force
```

# Configure PE 2.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#mpls router ldp
Ruijie(config-mpls-router)#ldp router-id interface loopback 0 force
```

Enable the LDP GR protocol, and configure parameters related to LDP GR.

# Configure PE 1.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#mpls router ldp
Ruijie(config-mpls-router)#graceful-restart
```

## Set the LDP reconnection time to 300 seconds, LDP neighbor keep-alive time to 120 seconds, and LDP recovery time to 120 seconds.

```
Ruijie(config-mpls-router)#graceful-restart timer reconnect 300
Ruijie(config-mpls-router)#graceful-restart timer neighbor-liveness 120
Ruijie(config-mpls-router)#graceful-restart timer recovery 120
Ruijie(config-mpls-router)#exit
```

# Configure the P device.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#mpls router ldp
Ruijie(config-mpls-router)#graceful-restart
```

## Set the LDP reconnection time to 300 seconds, LDP neighbor keep-alive time to 120 seconds, and LDP recovery time to 120 seconds.

```
Ruijie(config-mpls-router)#graceful-restart timer reconnect 300
Ruijie(config-mpls-router)#graceful-restart timer neighbor-liveness 120
Ruijie(config-mpls-router)#graceful-restart timer recovery 120
Ruijie(config-mpls-router)#exit
```

# Configure PE 2 by running the same commands as those on PE 1.

- Configure L3VPN.

# Configure PE 1.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
```

## Configure the eBGP peer CE.

```
Ruijie(config)#router bgp 200
Ruijie(config-router)#address-family ipv4 vrf 10
Ruijie(config-router-af)#neighbor 192.168.100.2 remote-as 100
Ruijie(config-router-af)#neighbor 192.168.100.2 update-source GigabitEthernet 2/1
Ruijie(config-router-af)#neighbor 192.168.200.2 activate
Ruijie(config-router-af)#exit-address-family
Ruijie(config-router)#exit
```

## Configure the iBGP peer PE 2.

```
Ruijie(config-router)#address-family ipv4
Ruijie(config-router-af)#neighbor 192.168.0.3 remote-as 200
Ruijie(config-router-af)#neighbor 192.168.0.3 update-source loopback 0
Ruijie(config-router-af)#neighbor 192.168.0.3 activate
Ruijie(config-router-af)#exit-address-family
Ruijie(config-router)#address-family vpnv4 unicast
Ruijie(config-router-af)#neighbor 192.168.0.3 activate
Ruijie(config-router-af)#exit-address-family
Ruijie(config-router)#exit
```

# Configure the P device. L3VPN configuration is not required on the P device.

# Configure PE 2 by running the same commands as those on PE 1.

- Enable the BGP GR protocol.

# Configure PE 1.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
```

## Enable the BGP and enter BPG configuration mode.

```
Ruijie(config)#router bgp 200
```

## Enable BGP GR.

```
Ruijie(config-router)#bgp graceful-restart
```

# Configure the P device. BGP GR does not need to be enabled on the P device.

# Configure PE 2 by running the same commands as those on PE 1.

Restart the LDP session for the configurations to take effect.

# Restart the LDP session on PE 1.

```
Ruijie#clear mpls ldp neighbor all
```

# Restart the LDP session on the PE device.

```
Ruijie#clear mpls ldp neighbor all
```

# Restart the LDP session on PE 1.

```
Ruijie#clear mpls ldp neighbor all
```

Verification

Run the following commands to view configurations on PE 1:

# View LDP GR information on PE 1.

```
Ruijie#show mpls ldp graceful-restart
Default VRF:
  LDP Graceful Restart is enabled
  Neighbor Liveness Timer: 120 seconds
  Max Recovery Time: 120 seconds
  Forwarding State Holding Time: 300 seconds
  Down Neighbor Database (1 records):
    Peer LDP Ident: 192.168.0.2:0; Local LDP Ident: 192.168.0.1:0
        Status: recovering (86 seconds left)
        Address list contains 3 addresses:
          192.168.0.2    192.168.1.2    192.168.2.1
  Graceful Restart-enabled Sessions:
   Peer LDP Ident: 192.168.0.2:0, State: estab
```

# View BGP GR information on PE 1.

```
Ruijie#show bgp vpnv4 unicast all neighbor
BGP neighbor is 192.168.0.3, remote AS 200, internal link
BGP version 4, remote router ID 192.168.0.3
BGP state = Established, up for 02:49:47
Last read 00:00:47, hold time is 180, keepalive interval is 60 seconds
Neighbor capabilities:
Route refresh: advertised and received(new)
Address family VPNv4 Unicast: advertised and received
Graceful Restart Capability: advertised and received
Remote Restart timer is 120 seconds
Address families preserved by peer:
VPNv4 Unicast
```

L2VPN GR

Note

L2VPN GR described in this section indicates GR for Virtual Pseudo Wire Service (VPWS) and Virtual Private LAN Service (VPLS).

Overview

For VPWS and VPLS services, public network tunnels are established based on basic MPLS network services. In addition, the extended LDP is used to distribute VC labels for the establishment of virtual lines. Therefore, extended LDP GR must be implemented to ensure uninterrupted forwarding of VPWS and VPLS services. Extended LDP GR is implemented based on the same working principles as basic LDP GR, except that the MPLS forwarding entries to be backed up are different. Therefore, extended LDP GR and basic LDP GR can be uniformly implemented.

Note

For details about LDP GR and configuration methods, see the "LDP GR" section in this document.
The working principles of VPWS GR are described in the following text. That of VPLS GR is not described in this document.

VPWS GR implements uninterrupted forwarding of VPWS services. It ensures that the data of VPWS services can be forwarded when the control plane on a device fails, protecting VPWS services on the network. The following prerequisites must be met for VPWS GR:

- Devices support 1+1 management board redundancy.
- Devices support uninterrupted forwarding of routing protocols.
- Devices support the LDP GR protocol.

The objectives of VPWS GR are to:

- Minimize routing protocol flapping during master/slave management board switchover.

- Minimize the impact on VPWS services.
- Minimize SPFs on access devices and improve VPWS network reliability.
- Minimize the packet loss rate of VPWS traffic.

Note

GR must be supported in unicast routing to implement uninterrupted forwarding of routing protocols. In other words, the device must support OSPF GR, IS-IS GR, or BGP GR.

Basic Concepts

GR Routers Classified by Capability

GR routers are classified by capability into GR-capable routers, GR-aware routers, and GR-unaware routers.

- GR-Capable Router

A GR-capable router is a router that has GR capability. In general, a GR-capable router is equipped with two management boards which work in 1+1 master/slave mode. The GR-capable router can send an advertisement packet to neighboring routers during master/slave switchover of the management boards, so that neighboring routers keep forwarding entries related to the GR-capable router. After master/slave switchover, routing tables are re-established without causing route flapping or changing the packet forwarding path, guaranteeing uninterrupted data forwarding in the system.

- GR-Aware Router

A GR-aware router is a router that has GR detection capability. It may be not equipped with two management boards but can detect that its neighbors are experiencing GR and can assist its neighbors to complete GR.

- GR-Unaware Router

A GR-unaware router is a router that does not have GR detection capability. It cannot detect that its neighbors are experiencing GR, cannot assist its neighbors to complete GR, and does not have GR capability. Generally, if a router has no awareness capability, it is because the system software does not provide the GR feature or the GR feature is disabled.

GR Routers Classified by Role

GR routers are classified by role during router restart into GR restarters and GR helpers.

- GR Restarter

The GR restarter has GR capability and its restart is triggered by administrators or faults.

- GR Helper

The GR helper is a neighbor of the GR restarter. It must be at least a GR-aware router.

Working Principle

Both the control plane and the forwarding plane on a traditional device are implemented by the same processor which simultaneously maintains a routing table and a forwarding table. To improve forwarding performance and reliability of devices, a multi-processor architecture is used on high-end and mid-range devices. The processors of control modules such as the routing protocol module are located on a master management board and data forwarding processors are located on line cards. In this manner,, the control plane and the forwarding plane are separated from each other so that data forwarding on line cards is not affected when the control plane is restarted. This technology provides a prerequisite for implementing GR. The GR-capable routers mentioned in this document are such routers where the control plane and the forwarding plane are separated from each other.

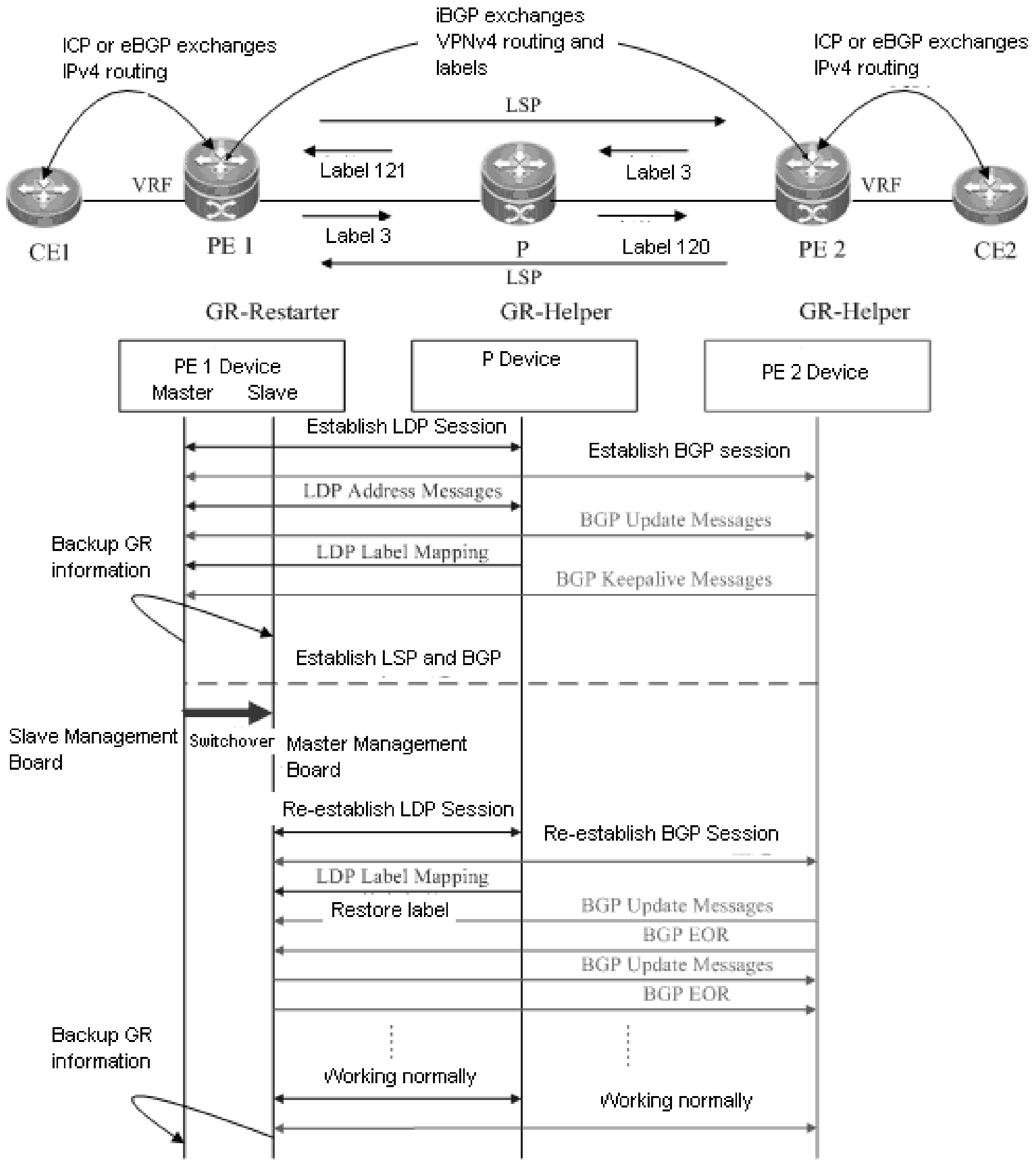
The VPWS network shown in Figure 1-5 has the following features:

- CE devices represent a customer network. The IGP runs on the CE devices.
- PE and P devices form a provider network. The IGP and LDP run on these devices.
- Public network tunnels and LSPs are established using the LDP between PE 1, PE 2, and the P device.
- Private network tunnels are established using the LDP between PE 1 and PE 2.
- The IGP and LDP have GR capability.
- PE devices are GR-capable routers, and the P device is a GR-aware router.

Note

For details about IGP GR and LDP GR, see related sections in *OSPF* and *LDP GR*.

Figure 1-5 VPWS networking topology



The following describes the GR process of the access device PE 1 of a provider network. Master/slave management board switchover occurs on PE 1. PE 1 works as a GR restarter, whereas PE 2 and the P device work as GR helpers. When master/slave management board switchover occurs on PE 1, a certain procedure applies. The procedure consists of the following three phases:

Before Master/Slave Management Board Switchover

PE 1 performs LDP GR and IGP GR negotiation with the P device, IGP GR negotiation with the connected CE device, and extended LDP GR negotiation with PE 2. PE 1 sends an Initialization message that carries the optional FT Session TLV to the P device and PE 2 to establish a GR-capable LDP session.

After the LDP session is established, they exchange LDP Address and Label Mapping messages with each other. In this manner, GR-capable LSPs are established for data forwarding.

When the master management board works normally, it must back up GR information to the slave management board to support GR.

During Master/Slave Management Board Switchover

GR information on PE 1 has been backed up to the slave management board. The major task of PE 1 in this phase is to perform master/slave management board switchover.

The P device and PE 2 detect that the TCP session is down, and therefore set an "old" flag for the respective LSPs. They also start a forwarding entry aging timer and continue to forward data before this aging timer expires. The working process on CEs is similar, except that it is an IGP GR process.

After Master/Slave Management Board Switchover

The slave management board on PE 1 becomes the new master management board, and the original master management board becomes the new slave management board. The new master management board starts to check the GR information previously backed up and determines whether forwarding entries before the restart are retained. Then the CLI configuration initialization process and the GR process continue. During LDP GR and IGP GR, all the devices send notification messages to all IGP and LDP neighbors for connection re-establishment.

- IGP convergence
- PE 1 sends an Initialization message that carries the FT Session TLV to the P device, and re-establishes a session upon receiving a response to obtain topology and routing information. Then PE 1 re-computes the routing table, deletes the "old" routes, and therefore completes IGP convergence.
- LDP processing
- PE 1 sends an Initialization message that carries the FT Session TLV to the neighboring P device and PE 2. A GR-capable LDP session is established after the P device and PE 2 receive the Initialization message. Then the devices exchange LDP Address and Label Mapping messages again with each other, and keep or remove the "old" flag set for MPLS forwarding entries. When the GR process ends, the P device, PE 1, and PE 2 delete the "old" MPLS forwarding entries from themselves respectively.

Note

The preceding IGP GR and LDP GR processes do not follow a strict priority sequence. In terms of route convergence, IGP routes converge first and converged routes are advertised to the LDP.

Before all protocols complete the GR process, only RIB information on the master management board is updated. The FIB information on interface boards do not be updated.

The FIB information on interface boards is updated only after all protocols complete the GR process.

Protocols and Specifications

The following protocols or specifications involved are as follows:

- RFC 3036: LDP Specification
- RFC 3037: LDP Applicability
- RFC 3215: LDP State Machine
- RFC 3478: Graceful Restart Mechanism for LDP
- RFC 3479: Fault Tolerance for LDP

Configuring VPWS GR

Network Environment

VPWS GR is required for VPWS applications on service bearing devices on an MPLS network. In this manner, data forwarding is not interrupted during master/slave management board switchover on devices, guaranteeing traffic continuity.

Note The GR capability does not guarantee traffic continuity when master/slave management board switchover also occurs on neighboring devices.

Prerequisites

Complete the following tasks before configuring VPWS GR:

- Build a VPWS environment and configure VPWS.
- Ensure that the IGP has GR capability.
- Ensure that the LDP has GR capability.

Note For details about VPWS configuration, see the "Configuring VPWS" section in *MPLS*.

Data Preparations

Prepare the following data before configuring VPWS GR:

- IGP GR parameters
- LDP GR parameters

Configuring IGP GR

Note For details about IGP GR configuration, see related sections in *OSPF*.

Configuring LDP GR

Note For details about LDP GR configuration, see the "LDP GR" section in this document.

Verification

To display VPWS GR configuration and running information, run the following commands:

Command	Function
show mpls ldp graceful-restart [all | vrf *vrf-name*]	Displays LDP GR sessions and session parameters.
show mpls ldp vc [all | vpws | hub | spoke] [*vc-id*]	Displays LDP Pseudo Wire (PW) information.

Note All the preceding commands can be configured in any mode except for the user mode.

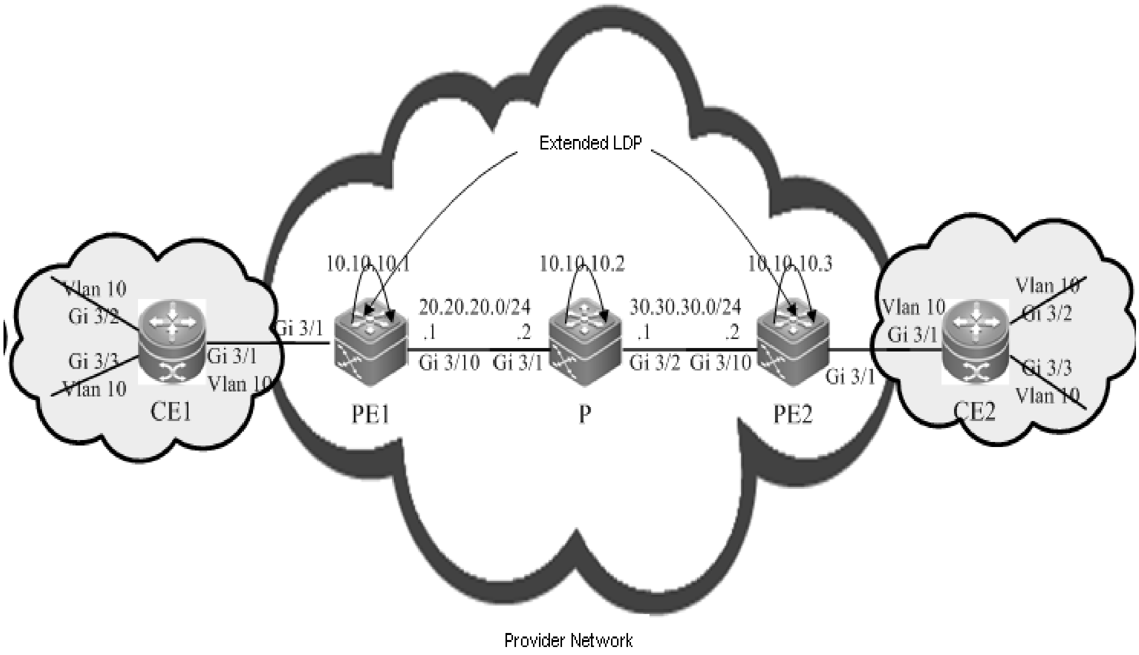
Configuration Examples

Networking Requirements

- Interconnection ports between PE and CE devices work in access mode, so that each CE device is connected to the respective PE device through an access link. The respective PE device establishes PW services for the Virtual Local Area Network (VLAN) to which the access port belongs. Since the Ethernet mode is applied, frames transmitted on the PW between PE 1 and PE 2 do not carry the VLAN 10 tag.
- CE, PE, and P devices support the LDP and are capable of GR.
- PE and P devices support the LDP and are capable of GR.
- PE and P devices form a provider network.
- PE devices are GR-capable routers, and the P device is a GR-aware router.

Networking Topology

Figure 1-6 Networking topology for configuring L2VPN GR



Configuration Tips

Configure PE 1, PE 2, and the P device as follows:

- Configure interface IP addresses and the OSPF protocol.
- Enable global MPLS forwarding on devices, MPLS packet forwarding, and the LDP on interfaces.
- Configure the LDP to enable the network to forward MPLS traffic.
- Configure VPWS.
- Enable the LDP GR protocol, and configure parameters related to LDP GR.
- Restart the LDP session for the configurations to take effect.

Configuration Steps

- Configure interface IP addresses and the OSPF protocol.

# Configure PE 1.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
```

## The **no switchport** command is used on a switch to switch to the Routed Port mode. It is not applicable to routers and therefore you do not need to run the command on routers.

```
Ruijie(config)#interface gigabitEthernet 3/10
Ruijie(config-if-GigabitEthernet 3/10)#no switchport
Ruijie(config-if-GigabitEthernet 3/10)#ip address 20.20.20.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 3/10)#exit
Ruijie(config)#interface loopback 0
Ruijie(config-Loopback 0)#ip address 10.10.10.1 255.255.255.255
Ruijie(config-Loopback 0)#exit
```

## Activate the OSPF protocol and enter OSPF mode.

```
Ruijie(config)#router ospf 10
Ruijie(config-router)#network 20.20.20.0 255.255.255.0 area 0
Ruijie(config-router)#network 10.10.10.1 255.255.255.255 area 0
Ruijie(config-router)#end
```

# Configure the P device.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
```

## The **no switchport** command is used on a switch to switch to Routed Port mode. It is not applicable to routers and therefore you do not need to run the command on routers.

```
Ruijie(config)#interface gigabitEthernet 3/1
Ruijie(config-if-GigabitEthernet 3/1)#no switchport
Ruijie(config-if-GigabitEthernet 3/1)#ip address 20.20.20.2 255.255.255.0
Ruijie(config-if-GigabitEthernet 3/1)#exit
```

## The **no switchport** command is used on a switch to switch to Routed Port mode. It is not applicable to routers and therefore you do not need to run the command on routers.

```
Ruijie(config)#interface gigabitEthernet 3/2
Ruijie(config-if-GigabitEthernet 3/2)#no switchport
Ruijie(config-if-GigabitEthernet 3/2)#ip address 30.30.30.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 3/2)#exit
```

## Configure the loopback interface **loopback 0**.

```
Ruijie(config)#interface loopback 0
Ruijie(config-Loopback 0)#ip address 10.10.10.1 255.255.255.255
Ruijie(config-Loopback 0)#exit
```

## Activate the OSPF protocol and enter OSPF mode.

```
Ruijie(config)#router ospf 10
Ruijie(config-router)#network 20.20.20.0 255.255.255.0 area 0
Ruijie(config-router)#network 30.30.30.0 255.255.255.0 area 0
Ruijie(config-router)#network 10.10.10.2 255.255.255.255 area 0
Ruijie(config-router)#end
```

# Configure PE 2.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
```

## The **no switchport** command is used on a switch to switch to Routed Port mode. It is not applicable to routers and therefore you do not need to run the command on routers.

```
Ruijie(config)#interface gigabitEthernet 3/10
Ruijie(config-if-GigabitEthernet 3/10)#no switchport
Ruijie(config-if-GigabitEthernet 3/10)#ip address 30.30.30.2 255.255.255.0
Ruijie(config-if-GigabitEthernet 3/10)#exit
```

## Configure the loopback interface **loopback 0**.

```
Ruijie(config)#interface loopback 0
Ruijie(config-Loopback 0)#ip address 10.10.10.3 255.255.255.255
Ruijie(config-Loopback 0)#exit
```

## Activate the OSPF protocol and enter OSPF mode.

```
Ruijie(config)#router ospf 10
Ruijie(config-router)#network 30.30.30.0 255.255.255.0 area 0
Ruijie(config-router)#network 10.10.10.3 255.255.255.255 area 0
Ruijie(config-router)#end
```

- Enable global MPLS forwarding on devices, MPLS packet forwarding, and the LDP on interfaces.

# Configure PE 1.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#mpls ip
```

## The **ip ref** command is used on a router to enable MPLS fast forwarding on the router. You do not need to run the command on switches.

```
Ruijie(config)#interface gigabitEthernet 3/10
Ruijie(config-if-GigabitEthernet 3/10)#label-switching
Ruijie(config-if-GigabitEthernet 3/10)#mpls ip
Ruijie(config-if-GigabitEthernet 3/10)#ip ref
Router(config-if-GigabitEthernet 3/10)#exit
```

# Configure the P device.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#mpls ip
```

## The **ip ref** command is used on a router to enable MPLS fast forwarding on the router. You do not need to run the command on switches.

```
Ruijie(config)#interface gigabitEthernet 3/1
Ruijie(config-if-GigabitEthernet 3/1)#label-switching
Ruijie(config-if-GigabitEthernet 3/1)#mpls ip
Ruijie(config-if-GigabitEthernet 3/1)#ip ref
Router(config-if-GigabitEthernet 3/1)#exit
```

## The **ip ref** command is run a router to enable MPLS fast forwarding on the router. You do not need to run this command on switches.

```
Ruijie(config)#interface gigabitEthernet 3/2
Ruijie(config-if-GigabitEthernet 3/2)#label-switching
Ruijie(config-if-GigabitEthernet 3/2)#mpls ip
Ruijie(config-if-GigabitEthernet 3/2)#ip ref
Router(config-if-GigabitEthernet 3/2)#exit
```

# Configure PE 2 by running the same commands as those on PE 1.

- Configure the LDP to enable the network to forward MPLS traffic.

# Configure PE 1.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#mpls router ldp
Ruijie(config-mpls-router)#ldp router-id interface loopback 0 force
```

## Configure a remote LDP neighbor.

```
Ruijie(config-mpls-router)#neighbor 10.10.10.3
Ruijie(config-mpls-router)#exit
```

# Configure the P device.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#mpls router ldp
Ruijie(config-mpls-router)#ldp router-id interface loopback 0 force
```

# Configure PE 2.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#mpls router ldp
Ruijie(config-mpls-router)#ldp router-id interface loopback 0 force
```

## Configure a remote LDP neighbor.

```
Ruijie(config-mpls-router)#neighbor 10.10.10.1
Ruijie(config-mpls-router)#exit
```

- Configure VPWS.

# Configure PE 1.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
```

## Switch configurations

### Configure access ports between PE 1 and CE 1.

```
Ruijie(config)#interface gigabitEthernet 3/1
Ruijie(config-if-GigabitEthernet 3/1)#switchport mode access
Ruijie(config-if-GigabitEthernet 3/1)#switchport access vlan 10
Ruijie(config-if-GigabitEthernet 3/1)#exit
```

### Configure PW services for VLAN 10 on PE 1.

```
Ruijie(config)#interface vlan 10
Ruijie(config-if-VLAN 10)#xconnect 10.10.10.3 2 encapsulation mpls ethernet
Ruijie(config-if-VLAN 10)#exit
```

## Router configurations

### Configure PW services for the GE interface 3/1 on PE 1.

```
Ruijie(config)#interface gigabitEthernet 3/1
Ruijie(config-if-GigabitEthernet 3/1)#ip ref
Ruijie(config-if-GigabitEthernet 3/1)#xconnect 10.10.10.3 2 encapsulation mpls ethernet
Ruijie(config-if-GigabitEthernet 3/1)#exit
```

# Configure PE 2.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
```

## Switch configurations

### Configure access ports between PE 2 and CE 2.

```
Ruijie(config)#interface gigabitEthernet 3/1
Ruijie(config-if-GigabitEthernet 3/1)#switchport mode access
Ruijie(config-if-GigabitEthernet 3/1)#switchport access vlan 10
Ruijie(config-if-GigabitEthernet 3/1)#exit
```

### Configure PW services for VLAN 10 on PE 2.

```
Ruijie(config-if-VLAN 10)#interface vlan 10
Ruijie(config-if-VLAN 10)#xconnect 10.10.10.1 2 encapsulation mpls ethernet
Ruijie(config-if-VLAN 10)#exit
```

## Router configurations

### Configure PW services for the GE interface 3/1 on PE 2.

```
Ruijie(config)#interface gigabitEthernet 3/1
Ruijie(config-if-GigabitEthernet 3/1)#ip ref
Ruijie(config-if-GigabitEthernet 3/1)#xconnect 10.10.10.1 2 encapsulation mpls ethernet
Ruijie(config-if-GigabitEthernet 3/1)#exit
```

- Enable the LDP GR protocol, and configure parameters related to LDP GR.

# Configure PE 1.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#mpls router ldp
Ruijie(config-mpls-router)#graceful-restart
```

## Set the LDP reconnection time to 300 seconds, LDP neighbor keep-alive time to 120 seconds, and LDP recovery time to 120 seconds.

```
Ruijie(config-mpls-router)#graceful-restart timer reconnect 300
Ruijie(config-mpls-router)#graceful-restart timer neighbor-liveness 120
Ruijie(config-mpls-router)#graceful-restart timer recovery 120
Ruijie(config-mpls-router)#exit
```

# Configure the P device.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#mpls router ldp
Ruijie(config-mpls-router)#graceful-restart
```

## Set the LDP reconnection time to 300 seconds, LDP neighbor keep-alive time to 120 seconds, and LDP recovery time to 120 seconds.

```
Ruijie(config-mpls-router)#graceful-restart timer reconnect 300
Ruijie(config-mpls-router)#graceful-restart timer neighbor-liveness 120
Ruijie(config-mpls-router)#graceful-restart timer recovery 120
Ruijie(config-mpls-router)#exit
```

# Configure PE 2 by running the same commands as those on PE 1.

- Restart the LDP session for the configurations to take effect.

# Restart the LDP session on PE 1.

```
Ruijie#clear mpls ldp neighbor all
```

# Restart the LDP session on the PE device.

```
Ruijie#clear mpls ldp neighbor all
```

# Restart the LDP session on PE 1.

```
Ruijie#clear mpls ldp neighbor all
```

Verification

Run the following commands to view configurations on PE 1:

# View LDP GR information on PE 1.

```
Ruijie#show mpls ldp graceful-restart
Default VRF:
  LDP Graceful Restart is enabled
  Neighbor Liveness Timer: 120 seconds
  Max Recovery Time: 120 seconds
  Forwarding State Holding Time: 300 seconds
  Down Neighbor Database (1 records):
    Peer LDP Ident: 10.10.10.2:0; Local LDP Ident: 10.10.10.1:0
        Status: recovering (86 seconds left)
        Address list contains 3 addresses:
          10.10.10.2   20.20.20.2   30.30.30.1
  Graceful Restart-enabled Sessions:
Peer LDP Ident: 10.10.10.2:0, State: estab
```

Configuring MPLS BFD

Note The routers or router icons involved in this chapter represent common routers or L3 switches where routing protocols are running.

Overview

Note MPLS Bidirectional Forwarding Detection (BFD) is implemented in accordance with *BFD For MPLS LSPs* as defined by IETF. As a component of BFD applications, MPLS BFD describes a method for detecting MPLS LSPs. For details about BFD, see related sections in *BFD*.

In general, the following methods are used to detect LSP faults on an MPLS network:

- MPLS OAM mechanism. It can effectively detect, confirm, and locate internal defects or faults on an MPLS network. Currently, the standardization for MPLS OAM is still under way and various OAM mechanisms are in a starting-off phase in terms of practical network applications. Therefore, not all devices on a network can support the OAM.
- Hello packet mechanism of the MPLS signaling protocol. It takes a long time for the mechanism to detect a fault. In general, the detection time is a matter of seconds. Therefore, plenty of traffic is lost if the hello mechanism is applied.

MPLS BFD can resolve all the preceding problems. It has the following features:

- MPLS BFD supports interworking and provides a unified detection mechanism for the entire network.
- MPLS BFD provides fast detection. It makes possible lightly-loaded fast detection to quicken the start of a backup forwarding path and therefore improves MPLS network reliability.
- MPLS BFD can detect MPLS LSP faults on the data plane. BFD uses a fixed packet format which facilitates hardware implementation and firewall transversal.

EFD Session Establishment

BFD uses a local discriminator My Discriminator and a remote discriminator Your Discriminator to differentiate BFD sessions between a pair of systems. The discriminators can be configured in either manual or auto mode.

- Manual configuration: The local discriminator and the remote discriminator are manually configured for BFD. In this manner, the LSP Ping Echo packet to be sent before BFD session establishment does not need to carry any discriminator to perform negotiation and learn the remote discriminator. Instead, a BFD session is directly established.

- Auto configuration: The LSP Ping Echo packet to be sent before BFD session establishment carries a discriminator to perform negotiation and learn the remote discriminator before a BFD session is established.

At the initial stage of BFD session establishment, the roles of devices at both ends are classified into initiators and passive LSRs. Whether the ingress/egress LSR is an initiator or a passive LSR depends on specific applications, but at least one of them must be the initiator. Therefore, the following two scenarios may exist at the initial stage:

Both Are Initiators

If both the ingress LSR and the egress LSR are initiators, LSPs are unidirectional. Therefore, this scenario can be further divided into the following two cases:

- BFD is applied to detect both LSPs from the ingress LSR to the egress LSR and from the egress LSR to the ingress LSR

The ingress LSR sends an LSP Ping echo request that carries a local discriminator to the egress LSR. When the egress LSR receives the echo request, it obtains a remote discriminator from the echo request, so that the egress LSR owns both a local discriminator generated by itself and a remote discriminator. Then the egress LSR sends a BFD control packet to the ingress LSR. When the ingress LSR receives the BFD control packet, it obtains a remote discriminator from the received BFD control packet. Therefore, the ingress LSR also owns a local discriminator generated by itself and a remote discriminator, and then sends a BFD control packet to the egress LSR. Till now, both LSRs proceed to the initial stage of BFD session establishment.

You must noted that the egress LSR may return or not return an echo reply upon receipt of the echo request. If the egress LSR returns an echo reply, the echo reply must carry a local discriminator generated by the egress LSR itself. In this manner, the ingress LSR can obtain a remote discriminator from either the BFD control packet or the echo reply.

Note The working process on the egress LSR is similar to that on the ingress LSR.

- BFD is applied to detect the LSP from the ingress LSR to the egress LSR, and detect IP addresses (in a multi-hop situation) from the egress LSR to the ingress LSR

In this case, discriminators are manually configured on the ingress LSR and the egress LSR to establish a BFD session. In other words, the BFD session is established without experiencing automatic discriminator negotiation but the session establishment process directly starts after discriminators are manually configured on the two LSRs.

One Is the Initiator and the Other Is a Passive LSR

The initiator sends an LSP Ping echo request that carries a local discriminator to the passive LSR. When the passive LSR receives the echo request, it obtains a remote discriminator from the echo request, so that it owns both a local discriminator generated by itself and a remote discriminator. Then the passive LSR sends a BFD control packet to the initiator. When the initiator receives the BFD control packet, it obtains a remote discriminator from the received BFD control packet. Therefore, the initiator also owns a local discriminator generated by itself and a remote discriminator, and then sends a BFD control packet to the passive LSR. Till now, both LSRs proceed to the initial stage of BFD session establishment.

You must noted that the passive LSR may return or not return an echo reply upon receipt of the echo request. If the passive LSR returns an echo reply, the echo reply must carry a local discriminator generated by the passive LSR itself. In this manner, the initiator can obtain a remote discriminator from either the BFD control packet or the echo reply.

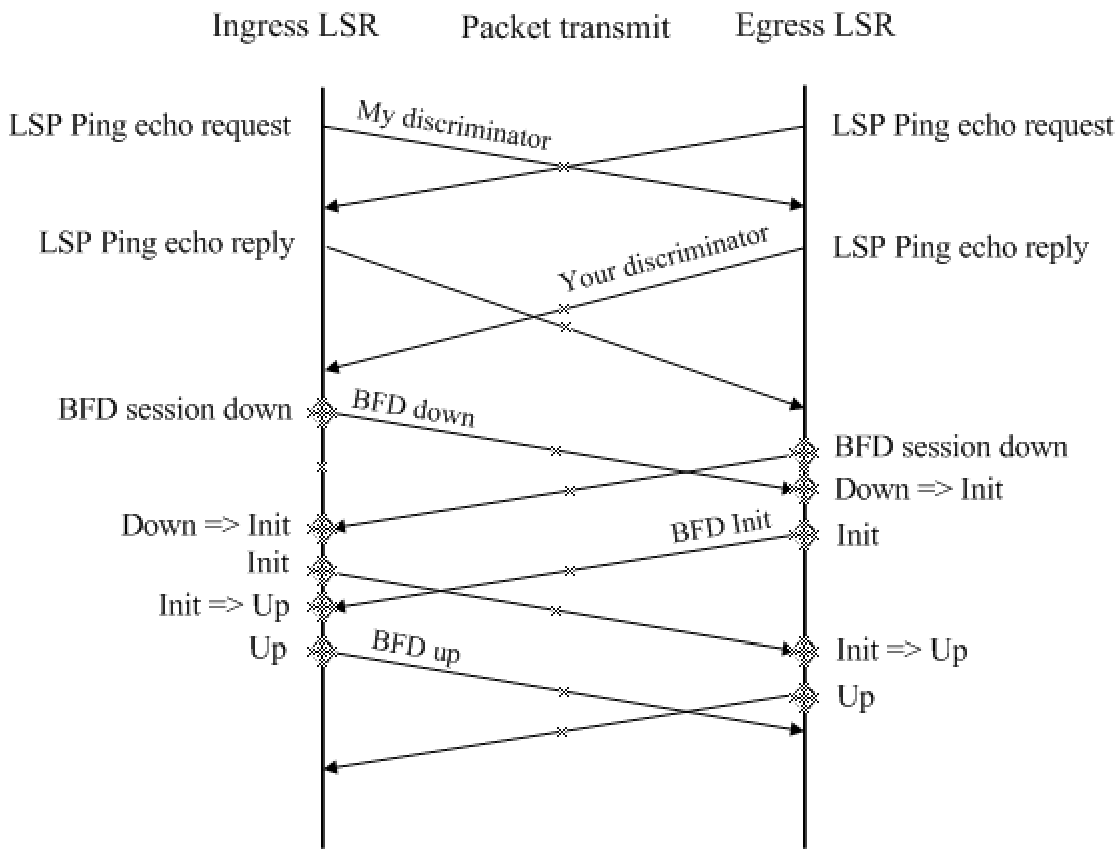
The passive LSR does not send any BFD control packet to the initiator unless it has received the echo request from the initiator.

Note Currently, only the first case is supported for BFD+LSP. That is, both LSRs must be initiators.

Example

This section describes the BFD session establishment process when both LSRs are initiators and BFD is applied to detect both LSPs from the ingress LSR to the egress LSR and from the egress LSR to the ingress LSR.

Figure 2-1 BFD session establishment process



- Before the ingress LSR and the egress LSR start BFD, they must learn the remote discriminator from each other and ensure that the LSPs are up. As shown in Figure 2-1, the ingress LSR sends an LSP Ping echo request that carries a local discriminator to the egress LSR. Upon receipt of the echo request, the egress LSR returns an echo reply that carries the local discriminator generated by the egress LSR itself to the ingress LSR. This is the same for the egress LSR. You must note that an LSR needs to learn the remote discriminator from the LSP Ping echo request if discriminators are not manually specified on both LSRs. If both the local discriminator and the remote discriminator are specified on both LSRs, this step does not apply during BFD session establishment but the next step directly continues. For details about discriminator configuration, see the "BFD+LSP" section in this document.

- The ingress LSR and the egress LSR start the BFD mechanism. The initial BFD status is Down on both LSRs. Each LSR sends a BFD packet that carries the Down status.
- Upon receipt of the BFD packet that carries the Down status, the egress LSR transits its local BFD status to "Init" and sends a BFD packet that carries the Init status.
- The egress LSR no longer processes any received BFD packet that carries the Down status after its local BFD status changes to the Init state.
- The same BFD status transition process applies on the ingress LSR.
- When receiving a BFD packet that carries the Init status, the egress LSR changes its local BFD status to the Up status.
- The same BFD status transition process applies on the ingress LSR.
- The status of the local BFD session status is Up, indicating that a BFD session has been successfully established.

BFD Modes

Before two LSRs exchange BFD control packets, a BFD session must be established on the condition that the control plane and the data plane take the same path. The following two modes exist for BFD session operations:

- Asynchronous mode
- Inquiry mode

In addition to the two operation modes, an echo function is defined for BFD. The echo function can be applied to both the asynchronous mode and the inquiry mode.

Note

Currently, only the asynchronous mode is supported for BFD+LSP. The inquiry mode and the echo function are not supported in the BFD+LSP scenario.

BFD+LSP

A BFD session is identified by a local discriminator My Discriminator and a remote discriminator Your Discriminator. The following two configuration modes can be applied to BFD+LSP based on the way by which the local discriminator and the remote discriminator are specified:

- Manual configuration
- In manual configuration mode, the local discriminator and remote discriminator are manually configured for BFD, so that the LSP Ping Echo packet to be sent before BFD session establishment does not need to carry any discriminator to perform negotiation and learn the remote discriminator. Instead, a BFD session is directly established.
- Auto configuration
- In automatic configuration mode, the LSP Ping Echo packet to be sent before BFD session establishment carries a discriminator to perform negotiation and learn the remote discriminator before a BFD session is established.

Currently, BFD+LSP is classified into BFD for static LSPs and BFD for LDP LSPs based on LSP type.

BFD for Static LSPs

BFD for static LSPs can be configured in manual or automatic mode. Since LSPs are unidirectional links but BFD is a bidirectional mechanism, reverse link detection can be performed based on one of the following detection methods when BFD is used to detect static LSPs:

- IP address mode for reverse link detection
- Static LSP mode for reverse link detection

BFD for LDP LSPs

BFD for static LSPs can be configured in manual or automatic mode. Since LSPs are unidirectional links but BFD is a bidirectional mechanism, reverse link detection can be performed based on one of the following detection methods when BFD is used to detect LDP LSPs:

- IP address mode for reverse link detection
- LDP LSP mode for reverse link detection

LDP LSPs bear basic VPN/PW public network services, so BFD for LDP LSPs is a mechanism for detecting faults of basic VPN/PW public network services. It provides fast detection for MPLS-based applications, such as VPN FRR and PW FRR, to protect services and guarantee MPLS network reliability.

Default Settings

Feature	Default
BFD session establishment mode	Active mode. Both LSRs must be initiators during BFD+LSP applications.
BFD mode	Asynchronous mode. Currently, only the asynchronous mode is supported for BFD+LSP. The inquiry mode and the echo function are not supported in the BFD+LSP scenario.
BFD session parameters	No default values are available. The BFD session parameters must be configured.
BFD authentication mode	Disabled. BFD authentication is not supported.

Protocols and Specifications

The following protocols or specifications involved are as follows:

- draft-ietf-bfd-base-09: Bidirectional Forwarding Detection
- draft-ietf-bfd-generic-05: Generic Application of BFD
- draft-ietf-bfd-mib-06: Bidirectional Forwarding Detection Management Information Base
- draft-ietf-bfd-v4v6-1hop-09: BFD for IPv4 and IPv6 (Single Hop)
- draft-ietf-bfd-multihop-07: BFD for IPv4 and IPv6 (Multihop)
- draft-ietf-bfd-mpls-07: BFD For MPLS LSPs

Configuring MPLS BFD

Configuring BFD for Static LSPs

Network Environment

BFD can be used to detect the continuity of static LSPs. When BFD is used for static LSPs, a static LSP that is down is not selected as the forwarding path of a private network route.

Prerequisites

Complete the following tasks before configuring BFD for static LSPs:

- Enable MPLS.
- Configure static LSPs.

Data Preparations

Prepare the following data before configuring BFD for static LSPs:

- My Discriminator and Your Discriminator of the BFD session
- Selection of the reverse link detection method
- BFD session parameters: BFD control packet sending interval, BFD control packet receiving interval, and the detection multiplier of BFD control packets

Configuring BFD on the Ingress LSR

By default, BFD for static LSPs is disabled on a device. To enable BFD for static LSPs on a device, enter privileged user mode to run the following commands:

Command	**Function**
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **interface type ID**	Enters the interface configuration mode.
Ruijie(config-if-type ID)# **bfd interval** *milliseconds* min_rx *milliseconds* multiplier *multiplier-value*	Sets BFD session parameters.
Ruijie(config-if-type ID)# **exit**	Exits interface configuration mode.
Ruijie(config)# **bfd bind static-lsp peer-ip** *ip-address* s**ource-ip** *ip-address* [**local-discriminator** *discr-value* **remote-discriminato**r *discr-value*] [**process-state**]	Sets BFD for static LSPs along with BFD session status handling. If manual configuration is applied on the ingress LSR, manual configuration must also be applied on the egress LSR. In other words, the configuration modes on both LSRs must be symmetrical.

To disable BFD for static LSPs, run the **no bfd bind static-lsp peer-ip** *ip-address* command.

Caution

Only static LSPs established by host route triggering are supported in BFD for static LSPs.

The *process-state* parameter must be specified for applications using BFD for fault detection, for example, when BFD is combined with LSP.

If discriminators are manually configured, the local and remote discriminators configured on the ingress LSR must match with those configured on the egress LSR.

Configuring BFD on the Egress LSR

By default, BFD for static LSPs is disabled on a device. To enable BFD for static LSPs on a device, enter privileged user mode to run the following commands:

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **interface type ID**	Enters interface configuration mode.
Ruijie(config-if-type ID)# **bfd interval** *milliseconds* min_rx *milliseconds* **multiplier** *multiplier-value*	Sets BFD session parameters.
Ruijie(config-if- type ID)# exit	Exits interface configuration mode.
Ruijie(config)# **bfd bind static-lsp peer-ip** *ip-address* **source-ip** *ip-address* [**local-discriminator** *discr-value* re**mote-discriminator** *discr-value*] **[process-state]**	Sets BFD for static LSPs without BFD session status handling. If manual configuration is applied on the ingress LSR, manual configuration must also be applied on the egress LSR. In other words, the configuration modes on both LSRs must be symmetrical.
Or:	
bfd bind backward-lsp-with-ip peer-ip *ip-address* **[vrf** *vrf-name*] **interface** *interface-type interface-number* **[source-ip** *ip-address*] {**local-discriminator** *discr-valu***e remote-discriminator** *discr-value*}	Sets the IP address mode for reverse link detection in BFD for LSPs. If the IP mode is applied for LSP reverse link detection during configuration, a local discriminator and a remote discriminator must be manually specified for the forward LSP.
Ruijie(config)# **exit**	Exits global configuration mode.

To disable BFD for static LSPs, run the **no bfd bind static-lsp peer-ip** *ip-address* or no bfd bind **backward-lsp-with-ip peer-ip** *ip-address* [**vrf** *vrf-name*] command.

Caution

Only static LSPs established by host route triggering are supported in BFD for static LSPs.

The IP detection mode can be applied on the reverse link during BFD for LSPs.

If discriminators are manually configured, the local and remote discriminators configured on the egress LSR must match with those configured on the ingress LSR.

Verification

To display configuration and running information about BFD for static LSPs, run the following commands:

Command	Function
show bfd neighbors [vrf *vrf-name*] [i**pv4** *ip-address* [**details] | client {static-lsp | backward-lsp-with-**ip} **[ipv4** *ip-address* [**details] | [details**]]	Displays BFD session information.

Configuring BFD for LDP LSPs

Network Environment

BFD can be used to detect the continuity of LDP LSPs. When BFD is used with LDP LSPs, an LDP LSP that is Down is not selected as the forwarding path of a private network route.

Prerequisites

Complete the following tasks before configuring BFD for LDP LSPs:

- Enable MPLS.
- Enable LDP.

Data Preparations

Prepare the following data before configuring BFD for LDP LSPs:

- My Discriminator and Your Discriminator of the BFD session
- Selection of the reverse link detection method
- BFD session parameters: BFD control packet sending interval, BFD control packet receiving interval, and the detection multiplier of BFD control packets

Configuring BFD on the Ingress LSR

By default, BFD for LDP LSPs is disabled on a device. To enable BFD for LDP LSPs on a device, enter privileged user mode to run the following commands:

Command	Function
Ruijie# **configure terminal**	Enters the global configuration mode
Ruijie(config)# **interface type ID**	Enters interface configuration mode.
Ruijie(config-if-type ID)# **bfd interval** *milliseconds* min_rx *milliseconds* **multiplier** *multiplier-value*	Sets BFD session parameters.
Ruijie(config-if-type ID)# **exit**	Exits interface configuration mode.
Ruijie(config)**#mpls router ldp**	Enters LDP configuration mode.
Ruijie(config-mpls-router)# **bfd bind ldp-lsp peer-ip** *ip-address* **nexthop** *ip-address* **[interface** *interface-type interface-number*] **source-ip** *ip-address* **[local-discriminator** *discr-value* **remote-discriminato**r *discr-value*] [**process-state**]	Sets BFD for LDP LSPs along with BFD session status handling. If manual configuration is applied on the ingress LSR, manual configuration must also be applied on the egress LSR. In other words, the configuration modes on both LSRs must be symmetrical.

To disable BFD for LDP LSPs, run the **no bfd bind ldp-lsp peer-ip** *ip-address* command.

Caution

Only LDP LSP established by host route triggering is supported in BFD for LDP LSPs.
One LSP can be bound to only one BFD session.
BFD binding can be performed only on the ingress LSR of the LDP LSP.
The *process-state* parameter must be specified for applications using BFD for fault detection, for example, when BFD is combined with LSP.
If discriminators are manually configured, the local and remote discriminators configured on the ingress LSR must match with those configured on the egress LSR.

Configuring BFD on the Egress LSR

By default, BFD for LDP LSPs is disabled on a device. To enable BFD for LDP LSPs on a device, enter the privileged user mode to run the following commands:

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **interface type ID**	Enters interface configuration mode.
Ruijie(config-if-type ID)# **bfd interval** *milliseconds* min_rx *millisecond*s **multiplier** *multiplier-value*	Sets BFD session parameters.
Ruijie(config-if-type ID)# **exit**	Exits interface configuration mode.
Ruijie(config)# **bfd bind backward-lsp-with-ip peer-ip** *ip-address* [**vrf** *vrf-name*] **interface** *interface-type interface-number* **[source-ip** *ip-address*] l**ocal-discriminator** *discr-value* **remote-discriminato**r *discr-value*	Sets the IP address mode for reverse link detection in BFD for LSPs. If the IP mode is applied for LSP reverse link detection during configuration, a local discriminator and a remote discriminator must be manually specified for the forward LSP.
Or:	
Ruijie(config)#**mpls router ldp**	Enters LDP configuration mode.
Ruijie(config-mpls-router)# **bfd bind ldp-lsp peer-i**p *ip-address* **nexthop** *ip-address* [**interface** *interface-type interface-number*] **source-ip** *ip-address* **[local-discriminator** *discr-value* **remote-discriminato**r *discr-value*] [**process-state**]	Sets BFD for LDP LSPs without BFD session status handling. If manual configuration is applied on the ingress LSR, manual configuration must also be applied on the egress LSR. In other words, the configuration modes on both LSRs must be symmetrical.
Ruijie(config-bfd-router)# **exit**	Exits the LDP configuration mode.

To disable BFD for LDP LSPs, run the **no bfd bind backward-lsp-with-ip peer-ip** *ip-address* [**vrf** *vrf-name*] or **no bfd bind ldp-lsp peer-ip** *ip-address* command.

Caution

The IP detection mode can be applied on the reverse link during BFD for LSPs.
Or BFD can be configured to detect the other LSP.
1. Only LDP LSPs established by host route triggering are supported in BFD for LDP LSPs.
2. One LSP can be bound to only one BFD session.

3. BFD binding can be performed only on the ingress LSR of the LDP LSP.

If discriminators are manually configured, the local and remote discriminators configured on the egress LSR must match with those configured on the ingress LSR.

Verification

To display configuration and running information about BFD for LDP LSPs, run the following commands:

Command	Function
show bfd neighbors [**vrf** *vrf-name*] [**ipv4** *ip-address* [**details] | client {ldp-lsp | backward-lsp-with-**ip} [**ipv4** *ip-address* [**details] | [details]**]	Displays BFD session information.

Configuration Examples

Configuring BFD for Static LSPs

Networking Requirements

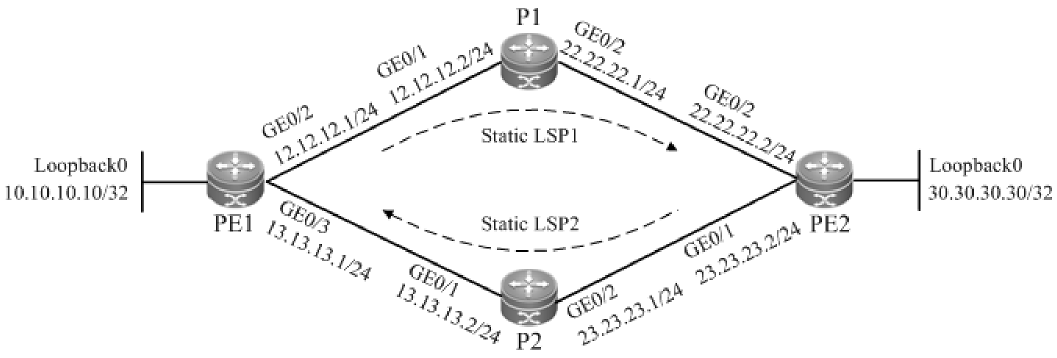
BFD can be configured to detect the continuity of static LSPs. Two links exist between PE 1 and PE 2, as shown in Figure 2-2.

- PE 1, PE 2, P1, and P2 form an MPLS network.
- A static LSP (LSP 1) exists and spans PE 1, P1 and PE 2 in turn. BFD is configured to detect this static LSP.
- A static LSP (LSP 2) exists and spans PE 2, P2 and PE 1 in turn. BFD is configured to detect this static LSP and notify faults if any to PE 1. This static LSP is used on the reverse link.

If LSP 1 fails, PE 1 can quickly receive a fault notification and handle the fault by deleting respective static MPLS routes.

Networking Topology

Figure 2-2 Networking topology for configuring BFD for static LSPs



Configuration Tips

Configure all devices as follows:

- Configure interface IP addresses and the OSPF protocol on devices.

- Enable global MPLS forwarding on all devices, and MPLS packet forwarding on interfaces.
- Configure static MPLS routes on the devices to enable the network to forward MPLS traffic.
- Configure BFD on PE 1 to detect static LSP 1.
- Configure BFD on PE 2 to detect static LSP 2, which is used as the LSP on the reverse link.

Configuration Steps

Configure interface IP addresses and the OSPF protocol on devices.

# Configure PE 1.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#interface gigabitEthernet 0/2
Ruijie(config-if-GigabitEthernet 0/2)#no switchport
Ruijie(config-if-GigabitEthernet 0/2)#ip address 12.12.12.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 0/2)#exit
Ruijie(config)#interface gigabitEthernet 0/3
Ruijie(config-if-GigabitEthernet 0/3)#no switchport
Ruijie(config-if-GigabitEthernet 0/3)#ip address 13.13.13.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 0/3)#exit
Ruijie(config)#interface loopback 0
Ruijie(config-Loopback 0)#ip address 10.10.10.10 255.255.255.255
Ruijie(config-Loopback 0)#exit
Ruijie(config)#router ospf 1
Router(config-router)#network 12.12.12.1 255.255.255.0 area 0
Router(config-router)#network 13.13.13.1 255.255.255.0 area 0
Router(config-router)#network 10.10.10.10 255.255.255.255 area 0
Router(config-router)#exit
```

# Configure the other devices by running the same commands as those on PE 1.

Enable global MPLS forwarding on all devices, and MPLS packet forwarding on interfaces.

# Configure PE 1.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#mpls ip
Ruijie(config)#interface gigabitEthernet 0/2
Ruijie(config-if-GigabitEthernet 0/2)#label-switching
Ruijie(config-if-GigabitEthernet 0/2)#mpls ip
Router(config-if-GigabitEthernet 0/2)#exit
Ruijie(config)#interface gigabitEthernet 0/3
Ruijie(config-if-GigabitEthernet 0/3)#label-switching
Ruijie(config-if-GigabitEthernet 0/3)#mpls ip
Router(config-if-GigabitEthernet 0/3)#exit
```

# Configure the other devices by running the same commands as those on PE 1.

Configure static MPLS routes on the devices to enable the network to forward MPLS traffic.

# Configure static LSP 1: PE1->P1->PE2.

# Configure PE 1.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#mpls static ftn 30.30.30.30/32 out-label 16 nexthop gigabitEthernet 0/2
12.12.12.2
```

# Configure P1.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#mpls static ilm in-label 16 forward-action swap-label 3 nexthop
gigabitEthernet 0/2 22.22.22.2 fec 30.30.30.30/32
```

## Run the **ping mpls ipv4** *30.30.30.30* command on PE 1 after the preceding configurations are complete. Ensure that the ping operation is successful.

# Configure static LSP 2: PE2->P2->PE1.

# Configure PE 2.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#mpls static ftn 10.10.10.10/32 out-label 16 nexthop gigabitEthernet 0/1
23.23.23.1
```

# Configure P2.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#mpls static ilm in-label 16 forward-action swap-label 3 nexthop
gigabitEthernet 0/1 13.13.13.1 fec 10.10.10.10/32
```

## Run the **ping mpls ipv4** *10.10.10.10* command on PE 2 after the preceding configurations are complete. Ensure that the ping operation is successful.

Configure BFD on PE 1 to detect LSP 1.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#interface gigabitEthernet 0/2
Ruijie(config-if-GigabitEthernet 0/2)#bfd interval 50 min_rx 50 multiplier 3
Ruijie(config-if-GigabitEthernet 0/2)#exit
Ruijie(config)#bfd bind static-lsp peer-ip 30.30.30.30 source-ip 10.10.10.10
local-discriminator 1 remote-discriminator 2 process-state
Ruijie(config)#exit
```

Configure BFD on PE 2 to detect LSP 2, which is used as the LSP on the reverse link.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#interface gigabitEthernet 0/1
Ruijie(config-if-GigabitEthernet 0/1)#bfd interval 50 min_rx 50 multiplier 3
Ruijie(config-if-GigabitEthernet 0/1)#exit
Ruijie(config)#bfd bind static-lsp peer-ip 10.10.10.10 source-ip 30.30.30.30
local-discriminator 2 remote-discriminator 1
Ruijie(config)#exit
```

Verification

View BFD session establishment information.

# View BFD session establishment information on PE 1.

```
Ruijie# show bfd neighbors details
OurAddr        NeighAddr      LD/RD  RH  Holdown(mult)   State   Int
10.10.10.10    30.30.30.30    1/2         1       532 (3 )       Up      Ge2/1
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 200000, MinRxInt: 200000, Multiplier: 3
Received MinRxInt: 50000, Received Multiplier: 3
Holdown (hits): 600(22), Hello (hits): 200(84453)
Rx Count: 49824, Rx Interval (ms) min/max/avg: 208/440/332
Tx Count: 84488, Tx Interval (ms) min/max/avg: 152/248/196
Registered protocols: static-lsp
Uptime: 02:18:49
Last packet:   Version: 1            - Diagnostic: 0
I Hear You bit: 1      - Demand bit: 0
Poll bit: 0            - Final bit: 0
Multiplier: 3             - Length: 24
My Discr.: 2           - Your Discr.: 1
Min tx interval: 50000    - Min rx interval: 50000
Min Echo interval: 0
```

# View BFD session establishment information on PE 2.

```
Ruijie# show bfd neighbors details
OurAddr        NeighAddr      LD/RD  RH  Holdown(mult)   State   Int
30.30.30.30    10.10.10.10    1/2         1       532 (3 )       Up      Ge2/1
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 200000, MinRxInt: 200000, Multiplier: 3
Received MinRxInt: 50000, Received Multiplier: 3
Holdown (hits): 600(22), Hello (hits): 200(84453)
Rx Count: 49824, Rx Interval (ms) min/max/avg: 208/440/332
Tx Count: 84488, Tx Interval (ms) min/max/avg: 152/248/196
Registered protocols: static-lsp
Uptime: 02:18:49
Last packet:   Version: 1            - Diagnostic: 0
```

```
I Hear You bit: 1     - Demand bit: 0
Poll bit: 0           - Final bit: 0
Multiplier: 3             - Length: 24
My Discr.: 2          - Your Discr.: 1
Min tx interval: 50000    - Min rx interval: 50000
Min Echo interval: 0
```

Configuring BFD for LDP LSPs

Networking Requirements

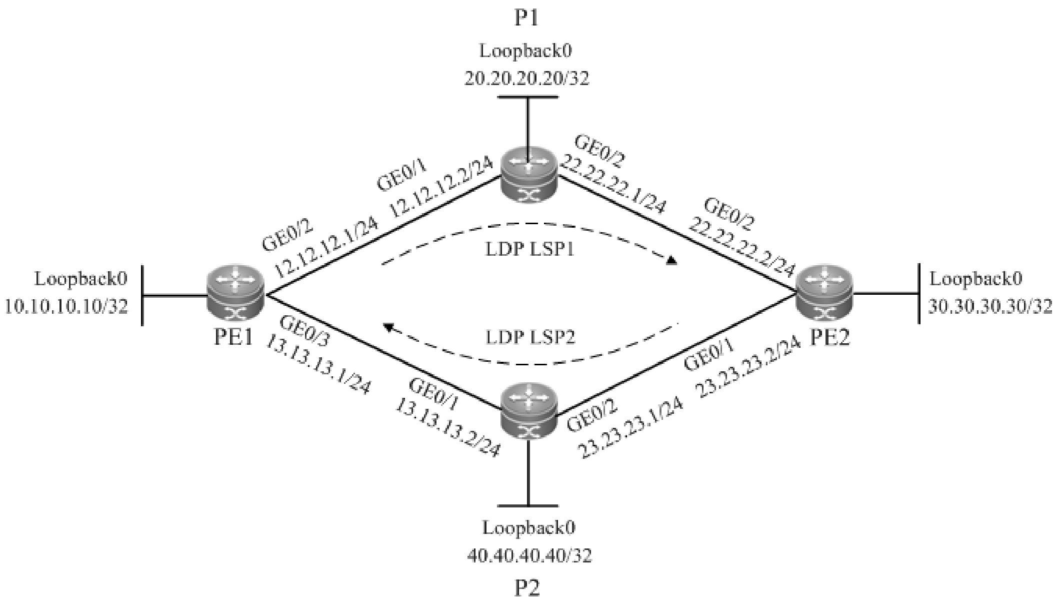
BFD can be configured to detect the continuity of LDP LSPs. Two links exist between PE 1 and PE 2, as shown in Figure 2-3.

- PE 1, PE 2, P1, and P2 form an MPLS network.
- Costs are configured for the interfaces on PE 1 and PE 2, so that two LSPs can be established between PE 1 and PE 2, as shown in Figure 2-3.
- An LDP LSP (LDP LSP 1) exists and spans PE 1, P1 and PE 2 in turn. BFD is configured to detect this LDP LSP.
- An LDP LSP (LDP LSP 2) exists and spans PE 2, P2 and PE 1 in turn. BFD is configured to detect this LDP LSP and notify faults if any to PE 1. This LDP LSP is used on the reverse link.

If LDP LSP 1 fails, PE 1 can quickly receive a fault notification and handle the fault by deleting respective MPLS routes.

Networking Topology

Figure 2-3 Networking topology for configuring BFD for LDP LSPs



Configuration Tips

Configure all devices as follows:

- Configure interface IP addresses and the OSPF protocol on devices.

- Enable global MPLS forwarding on devices, MPLS packet forwarding, and the LDP on interfaces.
- Configure the LDP to enable the network to forward MPLS traffic.
- Configure BFD on PE 1 to detect LDP LSP 1.
- Configure BFD on PE 2 to detect LDP LSP 2, which is used as the LSP on the reverse link.

Configuration Steps

Configure interface IP addresses and the OSPF protocol on devices.

# Configure PE 1.

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
```

## The **no switchport** command is used on a switch to switch to Routed Port mode. It is not applicable to routers, and therefore you do not need to run this command on routers.

```
Ruijie(config)#interface gigabitEthernet 0/2
Ruijie(config-if-GigabitEthernet 0/2)#no switchport
Ruijie(config-if-GigabitEthernet 0/2)#ip address 12.12.12.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 0/2)#exit
```

## The **no switchport** command is used on a switch to switch to Routed Port mode. It is not applicable to routers, and therefore you do not need to run this command on routers.

```
Ruijie(config)#interface gigabitEthernet 0/3
Ruijie(config-if-GigabitEthernet 0/3)#no switchport
Ruijie(config-if-GigabitEthernet 0/3)#ip address 13.13.13.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 0/3)#exit
Ruijie(config)#interface loopback 0
Ruijie(config-Loopback 0)#ip address 10.10.10.10 255.255.255.255
Ruijie(config-Loopback 0)#exit
Ruijie(config)#router ospf 1
Router(config-router)#network 12.12.12.1 255.255.255.0 area 0
Router(config-router)#network 13.13.13.1 255.255.255.0 area 0
Router(config-router)#network 10.10.10.10 255.255.255.255 area 0
Router(config-router)#exit
```

# Configure the other devices by running the same commands as those on PE 1.

Enable global MPLS forwarding on devices, MPLS packet forwarding, and the LDP on interfaces.

## Configure PE 1.

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)#mpls ip
```

## The **ip ref** command is run a router to enable MPLS fast forwarding on the router. You do not need to run this command on switches..

```
Ruijie(config)#interface gigabitEthernet 0/3
Ruijie(config-if-GigabitEthernet 0/3)#label-switching
Ruijie(config-if-GigabitEthernet 0/3)#mpls ip
Ruijie(config-if-GigabitEthernet 0/3)#ip ref
Router(config-if-GigabitEthernet 0/3)#exit
```

## The **ip ref** command is run a router to enable MPLS fast forwarding on the router. You do not need to run this command on switches.

```
Ruijie(config)#interface gigabitEthernet 0/3
Ruijie(config-if-GigabitEthernet 0/3)#label-switching
Ruijie(config-if-GigabitEthernet 0/3)#mpls ip
Ruijie(config-if-GigabitEthernet 0/3)#ip ref
Router(config-if-GigabitEthernet 0/3)#exit
```

# Configure P1.

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)#mpls ip
```

## The **ip ref** command is run a router to enable MPLS fast forwarding on the router. You do not need to run this command on switches.

```
Ruijie(config)#interface gigabitEthernet 0/1
Ruijie(config-if-GigabitEthernet 0/1)#label-switching
Ruijie(config-if-GigabitEthernet 0/1)#mpls ip
Ruijie(config-if-GigabitEthernet 0/1)#ip ref
Router(config-if-GigabitEthernet 0/1)#exit
```

## The **ip ref** command is run a router to enable MPLS fast forwarding on the router. You do not need to run this command on switches..

```
Ruijie(config)#interface gigabitEthernet 0/2
Ruijie(config-if-GigabitEthernet 0/2)#label-switching
Ruijie(config-if-GigabitEthernet 0/2)#mpls ip
Ruijie(config-if-GigabitEthernet 0/2)#ip ref
Router(config-if-GigabitEthernet 0/2)#exit
```

# Configure the other devices by running the same commands as those on PE 1.

Enable global MPLS forwarding on devices, MPLS packet forwarding, and the LDP on interfaces.

# Configure PE1

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)#mpls router ldp
Ruijie(config-mpls-router)#ldp router-id interface loopback 0 force
```

# Configure P1.

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)#mpls router ldp
Ruijie(config-mpls-router)#ldp router-id interface loopback 0 force
```

# Configure the other devices by running the same commands as those on PE 1.

Configure BFD on PE 1 to detect LDP LSP 1.

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)#interface gigabitEthernet 0/2
Ruijie(config-if-GigabitEthernet 0/2)#bfd interval 50 min_rx 50 multiplier 3
Ruijie(config-if-GigabitEthernet 0/2)#exit
Ruijie(config)#mpls router ldp
Ruijie(config-mpls-router)#bfd bind ldp-lsp peer-ip 30.30.30.30 nexthop 12.12.12.2
interface gigabitEthernet 0/2 source-ip 10.10.10.10 local-discriminator 1
remote-discriminator 2 process-state
Ruijie(config-mpls-router)#exit
```

Configure BFD on PE 2 to detect LDP LSP 2, which is used as the LSP on the reverse link.

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)#interface gigabitEthernet 0/1
Ruijie(config-if-GigabitEthernet 0/1)#bfd interval 50 min_rx 50 multiplier 3
Ruijie(config-if-GigabitEthernet 0/1)#exit
Ruijie(config)#mpls router ldp
Ruijie(config-mpls-router)#bfd bind ldp-lsp peer-ip 10.10.10.10 nexthop 23.23.23.1
interface gigabitEthernet 0/1 source-ip 30.30.30.30 local-discriminator 2
remote-discriminator 1
Ruijie(config-mpls-router)#exit
```

Verification

View BFD session establishment information.

# View BFD session establishment information on PE 1.

```
Ruijie# show bfd neighbors details
OurAddr        NeighAddr      LD/RD   RH  Holdown(mult)  State   Int
10.10.10.10    30.30.30.30    1/2        1      532 (3 )       Up     Ge2/1
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 200000, MinRxInt: 200000, Multiplier: 3
Received MinRxInt: 50000, Received Multiplier: 3
Holdown (hits): 600(22), Hello (hits): 200(84453)
Rx Count: 49824, Rx Interval (ms) min/max/avg: 208/440/332
Tx Count: 84488, Tx Interval (ms) min/max/avg: 152/248/196
Registered protocols: ldp-lsp
Uptime: 02:18:49
Last packet:   Version: 1            - Diagnostic: 0
I Hear You bit: 1     - Demand bit: 0
Poll bit: 0           - Final bit: 0
Multiplier: 3             - Length: 24
My Discr.: 2          - Your Discr.: 1
```

```
Min tx interval: 50000     - Min rx interval: 50000
Min Echo interval: 0
```

# View BFD session establishment information on PE 2.

```
Ruijie# show bfd neighbors details
OurAddr        NeighAddr      LD/RD   RH Holdown(mult)  State   Int
30.30.30.30    10.10.10.10    1/2        1       532 (3 )       Up      Ge2/1
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 200000, MinRxInt: 200000, Multiplier: 3
Received MinRxInt: 50000, Received Multiplier: 3
Holdown (hits): 600(22), Hello (hits): 200(84453)
Rx Count: 49824, Rx Interval (ms) min/max/avg: 208/440/332
Tx Count: 84488, Tx Interval (ms) min/max/avg: 152/248/196
Registered protocols: ldp-lsp
Uptime: 02:18:49
Last packet:   Version: 1             - Diagnostic: 0
I Hear You bit: 1      - Demand bit: 0
Poll bit: 0            - Final bit: 0
Multiplier: 3              - Length: 24
My Discr.: 2           - Your Discr.: 1
Min tx interval: 50000     - Min rx interval: 50000
Min Echo interval: 0
```

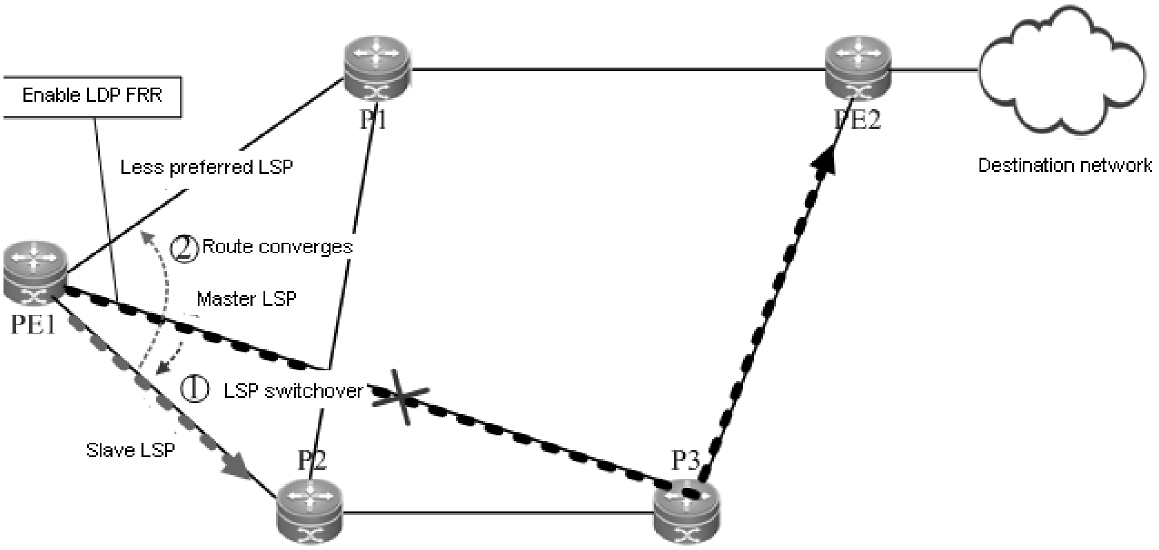
Configuring LDP FRR

Note The routers or router icons involved in this chapter represent common routers or L3 switches where routing protocols are running.

Overview

LDP Fast Rerouting (FRR) ensures that traffic can switch over from a master LSP to a backup LSP within a very short time when the master LSP fails and that traffic switches back to a new LSP from the backup LSP during route convergence. In this manner, traffic is not interrupted in the short time before network convergence, improving MPLS network reliability and protecting key services on the MPLS network.

Figure 3-1 Working principles of LDP FRR



LDP FRR is an extension of the LDP. The LDP works in free label retention mode. LDP FRR backs up retained labels (i.e. LSPs). When a link fails, the system can quickly detect the failure of the master LSP using a fast link failure detection technology such as BFD. While the LDP regenerates a new LSP, traffic on the master LSP switches back to the backup LSP to implement uninterrupted traffic forwarding. After the new LSP is generated, traffic is forwarded on the new LSP.

As shown in Figure 3-1. LDP FRR involves three types of LSPs: master LSP, less preferred LSP, and backup LSP. The master LSP is the optimal LSP along which traffic is forwarded when the network is stable and routes are converged. The less preferred LSP is one that has a cost value larger than the cost value of the master LSP. When the master LSP fails, routes converge on the less preferred LSP. The backup LSP is a backup LSP with a specified next hop. The three LSPs have different cost values.

In the LDP FRR solution, the next hop of the backup LSP interface is specified on the master LSP interface and BFD is configured on the master link to quickly detect faults of the master link. If the master link fails suddenly and a long time is required for routes to converge on the less preferred LSP, the quick link fault detection mechanism detects that the master link fails and therefore immediately switches traffic from the master LSP to the backup LSP, so that traffic is not discarded. In this manner, traffic to the destination network is forwarded by the backup LSP before routes converge on the less preferred LSP.

In the following several seconds, the master link fault is detected by the routing protocol and rerouting is performed through information exchange between routers. Finally, a notification message is sent to the LDP. The LDP regenerates a less preferred LSP based on the new next hop carried in the notification, and changes the path of the traffic to the destination network to the less preferred LSP, so that traffic smoothly switches over to the less preferred LSP from the backup LSP. If the less preferred LSP is just the backup LSP, traffic does not need to switch back to the less preferred LSP from the backup LSP. Even so, traffic still experiences the backup LSP phase and the less preferred LSP phase after route convergence, except that traffic just passes the same link in the two phases.

Configuring LDP FRR

Network Environment

When the LDP cannot effectively protect traffic on an MPLS network, LDP FRR can be configured on ports to protect the traffic and avoid traffic loss.

Prerequisites

Complete the following tasks before configuring LDP FRR:

- Enable MPLS.
- Configure MPLS LDP.

Data Preparations

Prepare the following data before configuring LDP FRR:

- Interface of the backup LSP, which is also the master interface
- Next-hop IP address of the backup LSP
- Name of the access control list (ACL)
- Priority of the backup LSP
- Length of the LDP FRR protection timer (optional)
- BRD single-hop detection parameters (optional)

Configuring the LDP FRR

By default, LDP FRR is disabled on a device. To enable the LDP FRR function on a device, enter the privileged user mode to run the following commands:

Command	**Function**
Ruijie#**configure terminal**	Enters global configuration mode.

Ruijie(config)#**mpls ip**	Enables MPLS forwarding. Caution This command is not applicable to forwarding on switch chips.
Ruijie(config)**#interface type ID**	Enters interface configuration mode.
Ruijie(config-if-type ID)#**mpls ip**	Enables LDP forwarding on interfaces (router configuration).
Ruijie(config-if-type ID)**#ip ref**	Enables fast forwarding on interfaces.
Ruijie(config-if-type ID)#**label-switching**	Enables MPLS packet processing on interfaces.
Ruijie(config-if-type ID)#**exit**	Exits the interface configuration mode.
Ruijie(config)**#mpls router ldp**	Enters the LDP configuration mode.
Ruijie(config-mpls-router)#**ldp router-id** *Loopback ID* **force**	Sets the router ID to a loopback ID (the settings immediately take effect).
Ruijie(config-mpls-router)#**exit**	Exits LDP configuration mode.
Ruijie(config)**#interface type ID**	Enters interface configuration mode.
Ruijie(config-if-type ID)# **mpls ldp frr nexthop** *nexthop-address* [**interface** *interface-type interface-number*] [**acl** *acl-name*] [**priority** *priority*]	Enables LDP FRR on interfaces.

To disable LDP FRR, run the **no mpls ldp frr** [**nexthop** *nexthop-address*] [**acl** *acl-name*] [**priority** *priority*] command.

Note Do not enable or disable LDP FRR during LDP GR.

Caution You must specify labels to work in free retention mode during LDP FRR configuration.

Configuring the LDP FRR Protection Timer (Optional)

If the master LSP link is recovered within the LDP FRR protection timer, traffic switches over to the master LSP link only after the LDP FRR protection timer expires. To configure the LDP FRR protection timer, enter the interface configuration mode to run the following commands in turn:

Command	**Function**
Ruijie**#configure terminal**	Enters global configuration mode.
Ruijie(config)**#mpls ip**	Enables MPLS forwarding. This command is not applicable to forwarding on switch chips.
Ruijie(config)#**interface type ID**	Enters the interface configuration mode.
Ruijie(config-if-type ID)**#mpls ip**	Enables LDP forwarding on interfaces.
Ruijie(config-if-type ID)#**ip ref**	Enables fast forwarding on interfaces.

Command	Function
Ruijie(config-if-type ID)#**label-switching**	Enables MPLS packet processing on interfaces.
Ruijie(config-if-type ID)#**exit**	Exits the interface configuration mode.
Ruijie(config)#**mpls router ldp**	Enters the LDP configuration mode.
Ruijie(config-mpls-router)#**ldp router-id** *Loopback ID* **force**	Sets the router ID to a loopback ID (the settings immediately take effect).
Ruijie(config-mpls-router)#**exit**	Exits the LDP configuration mode.
Ruijie(config)#**interface type ID**	Enters the interface configuration mode.
Ruijie(config-if-type ID)# **mpls ldp frr nexth**op *nexthop-address* [**acl** *acl-name*] [**priority** *priority*]	Enables LDP FRR on interfaces.
Ruijie(config-if-type ID)# **mpls ldp frr timer protect-time {infinity** | *seconds*}	Sets the LDP FRR protection timer on interfaces.

To enable the LDP FRR protection timer, run the **no mpls ldp frr timer protect-time** command.

Configuring Single-Hop BFD (Optional)

Run the **bfd bind peer-ip** command to configure single-hop BFD. After single-hop BFD is configured, the status of a BFD session is indicated in interface status information. Then LDP FRR detects the interface status and perform switchover. By default, single-hop BFD is disabled on a device. You can run the **no bfd bind peer-ip** command to disable single-hop BFD if necessary.

Note

Single-hop BFD is optional during LDP FRR configuration. For details about single-hop BFD configuration, see related sections in *BFD Configuration*.

Configuring DLDP (Optional)

Run the **dldp ip** command to configure the Device Link Detection Protocol (DLDP) function on a device. By default, probes are tried three times every 100 milliseconds. You can run the **no dldp ip** command to restore the default DLDP settings.

Note

For details about DLDP and configuration methods, see related sections in *DLDP*.

Verification

After configuring LDP FRR, run the following command to show LSP information:

Command	Function
show mpls rib	Displays MPLS routing information.

Configuration Examples

Networking Requirements

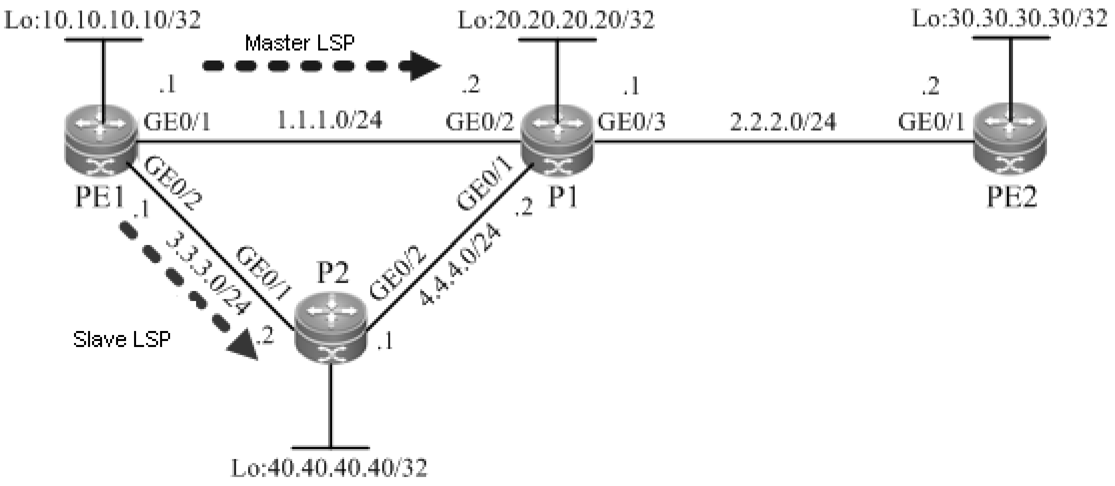
At least two links to the destination network are required when the LDP FRR is configured. Two links exist between PE 1 and PE 2, as shown in Figure 3-2.

- The PE1->P1->PE2 link is the master LSP.
- The PE1->P2->P1->PE2 link is the backup LSP.

LDP FRR can be configured on PE 1 and P1 to protect links between PE 1 and P1 and therefore avoid traffic loss.

Networking Topology

Figure 3-2 Networking topology for configuring LDP FRR



Configuration Tips

Configure all devices as follows:

- Configure interface IP addresses and the OSPF protocol on devices.
- Enable global MPLS forwarding on devices, MPLS packet forwarding, and the LDP on interfaces.
- Configure the LDP to enable the network to forward MPLS traffic.
- Configure LDP FRR.

116) Enable LDP FRR on the interfaces of PE 1 and P1 to generate the backup LSP.
117) Configure the LDP FRR protection timer on the interfaces of PE 1 and P1.
118) Configure single-hop BFD on the interfaces of PE 1 and P1.

Configuration Steps

- Configure interface IP addresses and the OSPF protocol on devices.

# Configure PE 1.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#interface gigabitEthernet 0/1
Ruijie(config-if-GigabitEthernet 0/1)#no switchport
Ruijie(config-if-GigabitEthernet 0/1)#ip address 1.1.1.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 0/1)#exit
Ruijie(config)#interface gigabitEthernet 0/2
Ruijie(config-if-GigabitEthernet 0/2)#no switchport
Ruijie(config-if-GigabitEthernet 0/2)#ip address 3.3.3.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 0/2)#exit
Ruijie(config)#interface loopback 0
Ruijie(config-Loopback 0)#ip address 10.10.10.10 255.255.255.255
Ruijie(config-Loopback 0)#exit
Ruijie(config)#router ospf 1
Router(config-router)#network 1.1.1.1 255.255.255.0 area 0
Router(config-router)#network 3.3.3.1 255.255.255.0 area 0
Router(config-router)#network 10.10.10.10 255.255.255.255 area 0
Router(config-router)#exit
```

# Configure the other devices by running the same commands as those on PE 1.

- Enable global MPLS forwarding on devices, MPLS packet forwarding, and the LDP on interfaces.

# Configure PE 1.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#mpls ip
Ruijie(config)#interface gigabitEthernet 0/1
Ruijie(config-if-GigabitEthernet 0/1)#label-switching
Ruijie(config-if-GigabitEthernet 0/1)#mpls ip
Router(config-if-GigabitEthernet 0/1)#exit
Ruijie(config)#interface gigabitEthernet 0/2
Ruijie(config-if-GigabitEthernet 0/2)#label-switching
Ruijie(config-if-GigabitEthernet 0/2)#mpls ip
Router(config-if-GigabitEthernet 0/2)#exit
```

# Configure the other devices by running the same commands as those on PE 1.

- Configure the LDP so that the network can forward MPLS traffic.

# Configure PE 1.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#mpls router ldp
Ruijie(config-mpls-router)#ldp router-id interface loopback 0 force
```

# Configure the other devices by running the same commands as those on PE 1.

- Configure LDP FRR.

# Configure PE 1.

## Enable LDP FRR on the interface of PE 1 to generate the backup LSP.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#interface gigabitEthernet 0/1
Ruijie(config-if-GigabitEthernet 0/1)#mpls ldp frr nexthop 3.3.3.2
Ruijie(config-if-GigabitEthernet 0/1)#exit
```

## Configure the LDP FRR protection timer on the interface of PE 1.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#interface gigabitEthernet 0/1
Ruijie(config-if-GigabitEthernet 0/1)#mpls ldp frr timer protect-time 15
Ruijie(config-if-GigabitEthernet 0/1)#exit
```

## Configure single-hop BFD on the interface of PE 1.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#interface gigabitEthernet 0/1
Ruijie(config-if-GigabitEthernet 0/1)#no bfd echo
Ruijie(config-if-GigabitEthernet 0/1)#bfd interval 50 min_rx 50 multiplier 3
Ruijie(config-if-GigabitEthernet 0/1)#bfd bind peer-ip 20.20.20.20 source-ip 1.1.1.1
process-pst
Ruijie(config-if-GigabitEthernet 0/1)#exit
```

# Configure P1.

## Enable LDP FRR on the interface of P1 so as to generate the backup LSP.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#interface gigabitEthernet 0/2
Ruijie(config-if-GigabitEthernet 0/2)#mpls ldp frr nexthop 4.4.4.1
Ruijie(config-if-GigabitEthernet 0/2)#exit
```

## Configure the LDP FRR protection timer on the interface of P1.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#interface gigabitEthernet 0/2
Ruijie(config-if-GigabitEthernet 0/2)#mpls ldp frr timer protect-time 15
Ruijie(config-if-GigabitEthernet 0/2)#exit
```

## Configure single-hop BFD on the interface of P1.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#interface gigabitEthernet 0/2
Ruijie(config-if-GigabitEthernet 0/2)#no bfd echo
Ruijie(config-if-GigabitEthernet 0/2)#bfd interval 50 min_rx 50 multiplier 3
Ruijie(config-if-GigabitEthernet 0/2)#bfd bind peer-ip 1.1.1.1 source-ip 1.1.1.2
process-pst
Ruijie(config-if-GigabitEthernet 0/2)#exit
```

Verification

View MPLS routing information.

# View MPLS routing information on PE 1.

```
Ruijie#show mpls rib
Status codes: m – main entry, b – backup entry, * - active, s – stale.
Default VRF:
LSP Information   Total
STATIC LSP       0
LDP LSP          2
RSVP LSP         0
BGP LSP          0
L3VPN LSP        0
LDP LSP:
-------------------------------------------------------------------------------------
----------------
FEC           In/Out Label    In/Out IF      Nexthop
m* 2.2.2.0/24     -/1024         -/Gi0/1        1.1.1.2
b  2.2.2.0/24     -/1025         -/Gi0/2        3.3.3.2
m* 30.30.30.30/32  -/1026         -/Gi0/1        1.1.1.2
b  30.30.30.30/32  -/1031         -/Gi0/2        3.3.3.2
-------------------------------------------------------------------------------------
----------------
```

Configuring LDP IGP Synchronization

Note

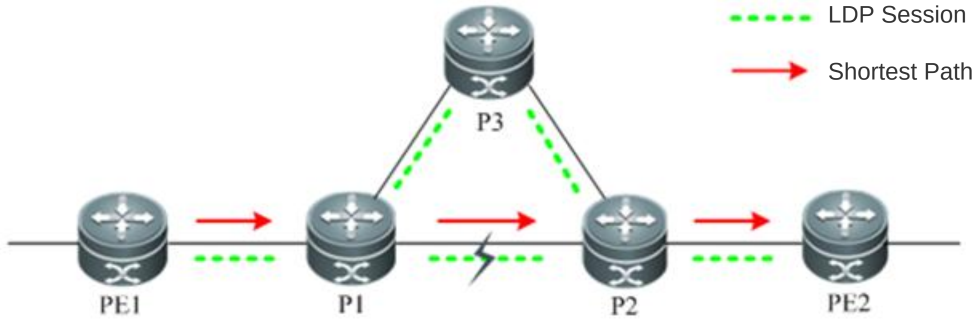
The routers and router icons described in this document represent those in common sense.

Overview

Link Detection Protocol (LDP)-Interior Gateway Protocol (IGP) synchronization is a technology that realizes uniform LDP and IGP convergence by changing the way of route release to avoid Multiple Protocol Label Switching (MPLS) route black holes.

In the MPLS network, LDP establishes Label Switched Paths (LSPs) based on the shortest IP path. As LDP and routing protocols have different convergence conditions and speeds, convergence of LDP LSPs may be inconsistent with that of the shortest IP path.

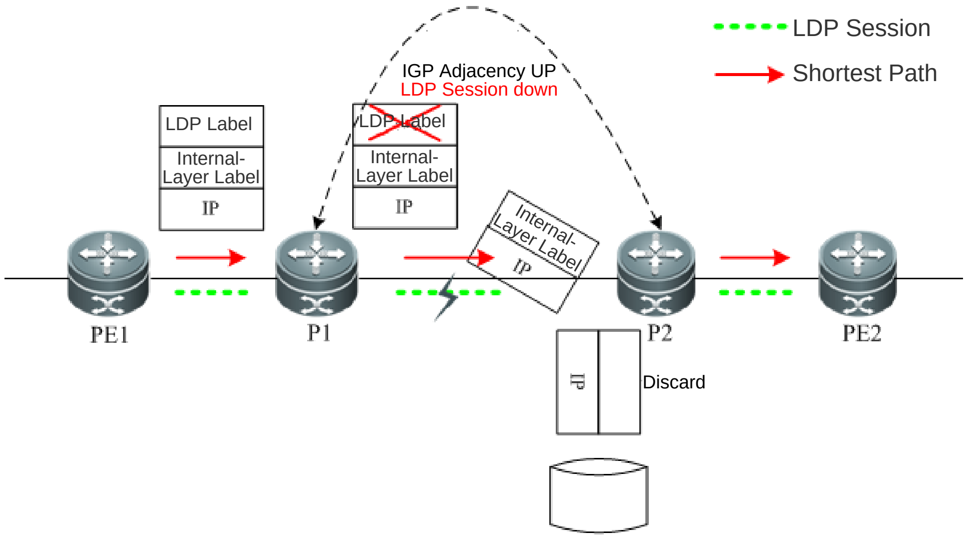
Figure 0-1 MPLS Network Topology



As shown in the figure, the devices are interconnected with each other at the link layer and run routing protocols and LDP. The shortest path from Provider Edge 1 (PE1) to PE2 is PE1-P1-P2-PE2, which covers LDP LSP. Routing protocols will not detect disconnection of the LDP session between P1 and P2 (the P1-P2 link and IGP remain normal) because the network topology is not changed. Although LDP detects the LSP disconnection, LDP cannot trigger route switching to a standby path because it is not a routing protocol. In this case, the shortest path from PE1 to PE2 is still PE1-P1-P2-PE2, but the LSP from PE1 to PE2 is disconnected between P1 and P2.

If MPLS Virtual Private Network (VPN)/6PE traffic is being forwarded through the path, the MPLS traffic will be affected.

Figure 0-2 MPLS VPN/6PE Traffic Being Forwarded through LSP

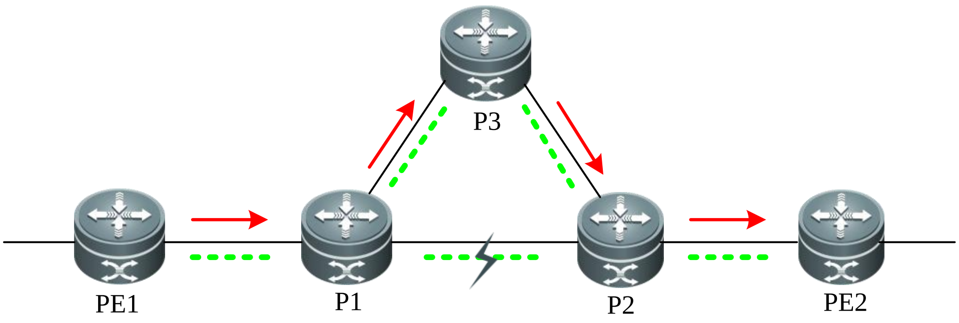


As shown in the figure above, the LSP between PE1 and PE2 is used to forward MPLS VPN/6PE traffic. Label packets are transmitted through the LDP LSP. When the LDP session between P1 and P2 is disconnected, packets have only inbound LDP labels on P1. P1 pops LDP labels upon receipt of the packets, and then forwards the packets with internal-layer labels to P2. However, P2 cannot identify internal-layer labels (because there is no private network route in the IPv4 routing table of the core device P on a public network and internal-layer labels are not assigned by P2), so the packets are discarded between P1 and P2, which results in traffic forwarding black holes.

Therefore, for applications such as MPLS VPN (including L2 VPN and L3 VPN) and 6PE that rely on end-to-end LDP LSPs on an MPLS network, if the convergence of LDP LSPs is inconsistent with that of the shortest IP path, MPLS route black holes will be incurred.

To solve this problem, you can configure LDP IGP synchronization on the core devices.

Figure 0-3 MPLS Network Topology



As shown in the figure, the LSP between PE1 and PE2 is used to forward MPLS VPN/6PE traffic. Label packets are transmitted through the LDP LSP. LDP IGP synchronization is enabled on core devices P1, P2 and P3. When the LDP session between P1 and P2 is disconnected, LDP IGP synchronization triggers IGP and LDP LSP convergence to the standby path P1-P3-P2, and the new path covers IGP and LDP LSP so that MPLS route black holes will not be incurred.

Caution You must to deploy standby paths for active links in the network topology; otherwise, it makes no sense to configure LDP IGP synchronization.

Configuring LDP IGP Synchronization

Scenario

LDP IGP synchronization aims at traffic loss caused by failure of the active LSP link on a network with active and standby LSP links. The following describes major application scenarios:

- When the active link is normal but the LDP session between two nodes on the active link fails, LSP traffic is switched to the standby link, while IGP traffic is still forwarded through the active link, which results in LSP traffic loss.
- When the active link fails, both IGP traffic and LSP traffic are switched to the standby link. However, when the fault is removed, as IGP converges faster than LDP, IGP is switched back to the active link before LDP does, which results in LSP traffic loss.

Prerequisites

Before configuring LDP IGP synchronization, you are advised to do the following:

- Enable MPLS.
- Configure MPLS LDP.

Data

Before configuring LDP IGP synchronization, you are advised to do the following:

- Configure IGP instances for LDP IGP synchronization (interfaces on which LDP IGP synchronization will be enabled).
- Configure the LDP IGP synchronization delay (optional).
- Configure the IGP waiting time for LDP synchronization (optional).

Enabling LDP IGP Synchronization

You can enable LDP IGP synchronization for all the interfaces that run Open Shortest Path First (OSPF) in OSPF routing process configuration mode. The following table describes the steps:

Command	Description
Ruijie # **configure terminal**	Enters global configuration mode.
Ruijie(config)# **router ospf** *process-id* [**vrf** *vrf-name*]	Creates an OSPF process and enters OSPF process configuration mode.
Ruijie(config-router)# **mpls ldp sync**	Enables LDP IGP synchronization.

Run the **no mpls ldp sync** command to disable LDP IGP synchronization.

(Optional) Enabling LDP IGP Synchronization in Interface Configuration Mode

You can enable/disable LDP IGP synchronization in interface configuration mode. By default, LDP IGP synchronization is enabled for all interfaces.

Command	Description
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **interface type ID**	Enters interface configuration mode.
Ruijie(config-if-type ID)#**mpls ldp igp sync**	Enables LDP IGP synchronization for an interface.

(Optional) Configuring a Timer for LDP IGP Synchronization Delay

You can configure the LDP IGP synchronization delay in interface configuration mode.

Command	Description
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **interface type ID**	Enters interface configuration mode.
Ruijie(config-if-type ID)#**mpls ldp igp sync delay** *30*	Configures the LDP IGP synchronization delay as 30 seconds.

(Optional) Configuring the IGP Waiting Time for LDP Synchronization

You can configure IGP waiting time for LDP synchronization in interface configuration mode.

Command	Description
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **interface type ID**	Enters interface configuration mode.
Ruijie(config-if-type ID)#**mpls ldp igp sync holddown***20*	Configures the IGP waiting time for LDP synchronization as 20 seconds.

Displaying Configuration Effect

You can run the following commands to display configurations.

Command	Description
show mpls ldp igp sync [**all | vrf** *vrf-name* **| interface** *interface-name*]	Displays configuration of LDP IGP synchronization.
show ip ospf [*process-id*] **mpls ldp interface** [*interface-name*]	Displays configuration of LDP IGP synchronization for an OSPF-enabled interface.

Configuration Example

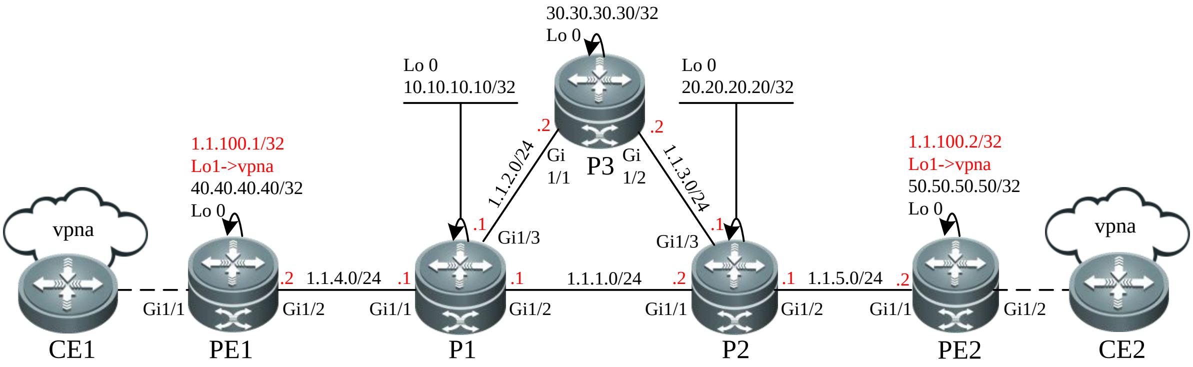
OSPF Network

Networking Requirements

The public network of an Internet Service Provider (ISP) carries VPN traffic. Each device runs OSPF. To improve the network reliability and avoid route black holes caused by inconsistent LDP and OSPF convergence, you can configure LDP IGP synchronization on the core device P.

Network Topology

Figure 0-4 OSPF Network Topology for Configuring LDP IGP Synchronization



Configuration Tips

The following describes Configuration Tips:

- Configure interface IP addresses on each node, configure OSPF and enable LDP.
- Configure VPN route instances and MP-IBGP neighbors and transfer VPN routing information on PEs.
- Enable LDP IGP synchronization on P1, P2 and P3.

Steps

- Configure interface IP addresses on each node, configure OSPF and enable LDP.

# Configure PE1.

## Configure a loopback interface.

```
Ruijie(config)# interface Loopback 0
Ruijie(config-if-Loopback 0)# ip address 40.40.40.40 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

## Enable MPLS and LDP in global configuration mode.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface Loopback 0force
Ruijie(config-mpls-router)# exit
```

## Configure interface IP addresses and enable MPLS and LDP for the interfaces.

```
Ruijie(config)# interface gigabitEthernet 1/2
Ruijie(config-if-GigabitEthernet 1/2)# ip address1.1.4.2 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/2)# mpls ip
Ruijie(config-if-GigabitEthernet 1/2)# label-switching
Ruijie(config-if-GigabitEthernet 1/2)# exit
```

## Configure IGP (OSPF).

```
Ruijie(config)# router ospf 1
Ruijie(config-router)#network1.1.4.00.0.0.255 area 0
Ruijie(config-router)#network 40.40.40.400.0.0.0 area 0
Ruijie(config-router)# exit
```

# Configure P1.

## Configure a loopback interface.

```
Ruijie(config)# interface Loopback 0
Ruijie(config-if-Loopback 0)# ip address 10.10.10.10 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

## Enable MPLS and LDP in global configuration mode.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface Loopback 0force
Ruijie(config-mpls-router)# exit
```

## Configure interface IP addresses and enable MPLS and LDP for the interfaces.

```
Ruijie(config)# interface gigabitEthernet 1/1
Ruijie(config-if-GigabitEthernet 1/1)# ip address1.1.4.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/1)# mpls ip
Ruijie(config-if-GigabitEthernet 1/1)# label-switching
Ruijie(config-if-GigabitEthernet 1/1)# exit
Ruijie(config)# interface gigabitEthernet 1/2
Ruijie(config-if-GigabitEthernet 1/2)# ip address1.1.1.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/2)# mpls ip
Ruijie(config-if-GigabitEthernet 1/2)# label-switching
Ruijie(config-if-GigabitEthernet 1/2)# exit
Ruijie(config)# interface gigabitEthernet 1/3
Ruijie(config-if-GigabitEthernet 1/3)# ip address1.1.2.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/3)# mpls ip
Ruijie(config-if-GigabitEthernet 1/3)# label-switching
Ruijie(config-if-GigabitEthernet 1/3)# exit
```

## Configure IGP (OSPF).

```
Ruijie(config)# router ospf 1
Ruijie(config-router)#network1.1.1.00.0.0.255 area 0
Ruijie(config-router)#network1.1.2.00.0.0.255 area 0
Ruijie(config-router)#network1.1.4.00.0.0.255 area 0
Ruijie(config-router)#network10.10.10.100.0.0.0 area 0
Ruijie(config-router)#exit
```

# Configure P3.

## Configure a loopback interface.

```
Ruijie(config)# interface Loopback 0
Ruijie(config-if-Loopback 0)# ip address 30.30.30.30 255.255.255.255
Ruijie(config-if-Loopback 0)# exit
```

## Enable MPLS and LDP in global configuration mode.

```
Ruijie(config)# mpls ip
Ruijie(config)# mpls router ldp
Ruijie(config-mpls-router)# ldp router-id interface Loopback 0force
Ruijie(config-mpls-router)# exit
```

## Configure interface IP addresses and enable MPLS and LDP for the interfaces.

```
Ruijie(config)# interface gigabitEthernet 1/1
Ruijie(config-if-GigabitEthernet 1/1)# ip address1.1.2.2 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/1)# mpls ip
Ruijie(config-if-GigabitEthernet 1/1)# label-switching
Ruijie(config-if-GigabitEthernet 1/1)# exit
Ruijie(config)# interface gigabitEthernet 1/2
Ruijie(config-if-GigabitEthernet 1/2)# ip address1.1.3.2 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/2)# mpls ip
Ruijie(config-if-GigabitEthernet 1/2)# label-switching
Ruijie(config-if-GigabitEthernet 1/2)# exit
```

## Configure IGP (OSPF).

```
Ruijie(config)# router ospf 1
Ruijie(config-router)#network1.1.2.00.0.0.255 area 0
Ruijie(config-router)#network1.1.3.00.0.0.255 area 0
Ruijie(config-router)#network 30.30.30.300.0.0.0 area 0
Ruijie(config-router)#exit
```

# Configuration of P2/PE2 is the same as that of P1/PE1, respectively.

- Configure VPN routing instances and MP-IBGP neighbors and transfer VPN routing information on PEs.

## Configure VPN Routing & Forwarding (VRF): Take PE1 as an example, which is similar to PE2.

```
Ruijie(config)# ip vrf vpna
Ruijie(config-vrf)# rd 65001:20
Ruijie(config-vrf)# route-target both 65001:20
Ruijie(config-vrf)# exit
Ruijie(config)# interface gigabitEthernet 1/1
Ruijie(config-if-GigabitEthernet 1/1)# ip vrf forwarding vpna
Ruijie(config-if-GigabitEthernet 1/1)# ip address 1.1.100.1 255.255.255.0
```

## Configure MP-IBGP neighbors and transfer VPN routing instances: Take PE1 as an example, which is similar to PE2.

```
Ruijie(config)# router bgp 65001
Ruijie(config-router)# neighbor 50.50.50.50 remote-as 65001
Ruijie(config-router)# neighbor 50.50.50.50 update-source Loopback 0
Ruijie(config-router)# address-familyvpnv4
Ruijie(config-router-af)# neighbor 50.50.50.50 activate
Ruijie(config-router-af)# neighbor 50.50.50.50 send-community both
Ruijie(config-router-af)# exit
Ruijie(config-router)# address-family ipv4 vrf vpna
Ruijie(config-router-af)# redistribute connected
Ruijie(config-router-af)# exit
Ruijie(config-router)# exit
```

- Enable LDP IGP synchronization on P1, P2 and P3.

# Enable LDP IGP synchronization: Take P1 as an example, which is similar to P2 and P3.

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)#router ospf 1
Ruijie(config-router)#mpls ldp sync
Ruijie(config-router)#exit
```

Verification

Display whether LDP IGP synchronization is enabled.

# Display whether LDP IGP synchronization is enabled on P1.

```
Ruijie# show mpls ldp igp sync
   GigabitEthernet 1/1:
       LDP configured; LDP-IGP Synchronization enabled.
       Sync status: sync required; not achieved.
       Sync delay time: 5 seconds
       IGP holddown time: infinite.
       Peer LDP Ident: 40.40.40.40:0
       IGP enabled: OSPF 1
   GigabitEthernet 1/2:
       LDP configured; LDP-IGP Synchronization enabled.
       Sync status: sync required; not achieved.
       Sync delay time: 5 seconds
       IGP holddown time: infinite.
       Peer LDP Ident: 20.20.20.20:0
       IGP enabled: OSPF 1
   GigabitEthernet 1/3:
       LDP configured; LDP-IGP Synchronization enabled.
       Sync status: sync required; not achieved.
       Sync delay time: 5 seconds
       IGP holddown time: infinite.
       Peer LDP Ident: 30.30.30.30:0
       IGP enabled: OSPF 1
```

Configuring MPLS ECMP

Note The routers and router icons described in this document represent those in common sense.

Overview

Establishment of Multiple Protocol Label Switching (MPLS) forwarding entries is driven by IP routing. If a Forward Error Correction (FEC) has multiple equal cost next hops, the MPLS signaling protocol can establish multiple equal cost next hops for the FEC, and the FEC traffic can be forwarded through the equal cost paths. When an MPLS network supports Equal-Cost MultiPath (ECMP), Facsimile Transmission Network (FTN) or Identity Lifecycle Manager (ILM) for the same FEC can be mapped to multiple Next Hop Label Forwarding Entries (NHLFEs), and each next hop matches one NHLFE. The FEC packets can be forwarded through the equal cost paths, thereby achieving load balancing.

The following describes the benefits of load balancing:

- Increases the MPLS network bandwidth without updating hardware.
- Strengthens protection of MPLS network links (end-to-end and point-to-point). Balances traffic to multiple links. When one of these links fails, traffic can be forwarded through other links, which improves reliability.
- Balances MPLS traffic toward the same destination on multiple Label Switched Paths (LSPs) and thereby makes the most of network resources.

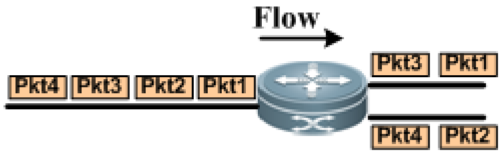
Working Principle

Scheduling Algorithms for MPLS Load Balancing

Scheduling algorithms for MPLS load balancing aim to select a next hop for an FEC packet from multiple equal cost paths. There are many algorithms, among which per packet scheduling and per flow scheduling algorithms are frequently used.

Per Packet Scheduling

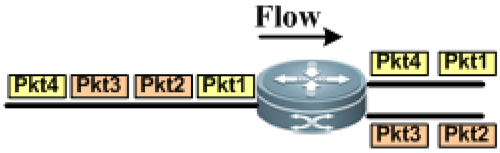
Figure 0-1 Per Packet Scheduling



Per packet scheduling forwards the packets toward the same destination network successively to different equal cost next-hop links to balance the traffic equally. As shown in the following figure, on a Link-State Request (LSR), two equal cost links are destined for the same destination network. There are 4 packets. Packet 1 is transmitted through Link 1, Packet 2 through Link 2, Packet 3 through Link 1, and Packet 4 through Link 2. This is called per packet scheduling.

Per Flow Scheduling

Figure 0-2 Per Flow Scheduling



Under per flow scheduling, the router checks certain fields of packets and uses these fields as key words to map the packets of the same type to the same next-hop link through a hash function. As shown in the figure, there are 4 packets destined for the same destination network. Packet 1 and Packet 4 are of the same type; Packet 2 and Packet 3 are of the same type. Packet 1 and Packet 4 are forwarded to Link 1, and Packet 2 and Packet 3 are forwarded to Link 2 by hashing the fields that mark the packet type. This is called per flow scheduling.

Selection of hash key words:

- MPLS LSP load balancing

  If the MPLS label stack is followed by the IPv4 header, use destination and source IP addresses for hash calculation and select a next hop for packet forwarding. If the MPLS label stack is not followed by the IPv4 header, use the innermost label for hash calculation and select a next hop for packet forwarding; otherwise, hash calculation is not performed and packets are forwarded to the first valid next hop.
- MPLS L3VPN load balancing

  Use destination and source IP addresses for hash calculation and select a next hop to forward packets.
- MPLS Virtual Private Wire Service (VPWS) load balancing

  Use Pseudo Wire (PW) labels for hash calculation and select a next hop to forward packets.
- MPLS Virtual Private LAN Service (VPLS) load balancing

  Use destination and source MAC addresses for hash calculation and select a next hop to forward packets.

Protocols and Standards

N/A.

Defaults

Feature	Default Settings
Load balancing mode	Per flow scheduling is used.
Establish equal cost LSPs for all public network routes	Equal cost LSPs are established only for host routes.

Configuring MPLS ECMP

Establishing Equal Cost LSPs for All Equal Cost Routing FECs

Command	Description

Command	Description
Ruijie#**configure terminal**	Enters global configuration mode.
Ruijie(config)#**mpls router ldp**	Enables MPLS forwarding.
Ruijie(config-mpls-router)# **ldp-ecmp for all-fec**	Establishes equal cost LSPs for all equal cost routing FECs.

Configuring MPLS Load Balancing Mode

Command	Description
Ruijie#**configure terminal**	Enters global configuration mode.
Ruijie(config)# **mpls load-balance** { **flow| packet** }	Configures the MPLS load balancing mode.

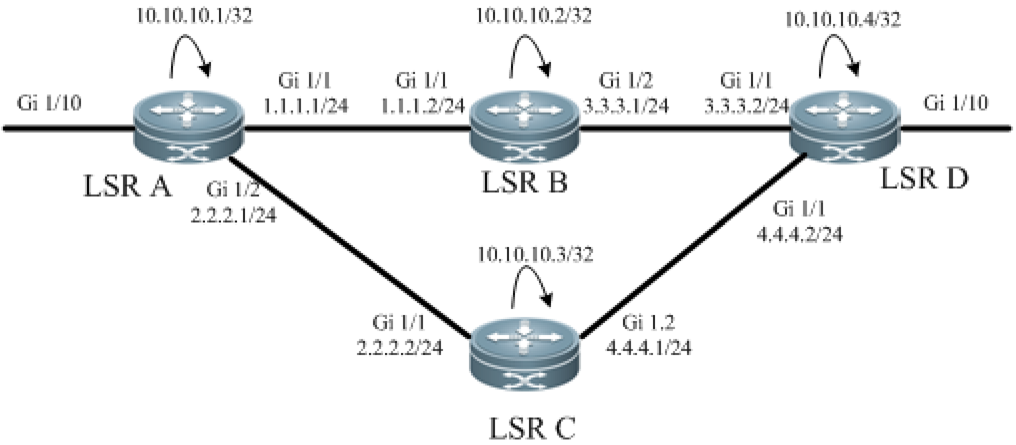
Configuration Example

Networking Requirements

As shown in the following figure, all nodes support MPLS and run Open Shortest Path First (OSPF) as the Interior Gateway Protocol (IGP) on the MPLS backbone network. Two LSPs from LSR A to LSR D are available for load balancing: LSR A→LSR B→LSR D and LSR A→LSR C→LSR D.

Network Topology

Figure 0-3 Network Topology for MPLS Load Balancing



Configuration Tips

The following describes the Configuration Tips of configuration for each node:

Configure the interface IP address and OSPF protocol.

Configure MPLS and MPLS Link Detection Protocol (LDP) in global and interface configuration modes.

Configure the load balancing policy on the start node.

Steps

Configure the interface IP address and OSPF protocol.

# Configure LSR A.

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)#interface gigabitEthernet 1/1
Ruijie(config-if-GigabitEthernet 1/1)#no switchport
Ruijie(config-if-GigabitEthernet 1/1)#ip address 1.1.1.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/1)#exit
Ruijie(config)#interface gigabitEthernet 1/2
Ruijie(config-if-GigabitEthernet 1/2)#no switchport
Ruijie(config-if-GigabitEthernet 1/2)#ip address 2.2.2.1 255.255.255.0
Ruijie(config-if-GigabitEthernet 1/2)#exit
Ruijie(config)#interface loopback 0
Ruijie(config-Loopback 0)#ip address 10.10.10.1 255.255.255.255
Ruijie(config-Loopback 0)#exit
Ruijie(config)#router ospf 1
Ruijie(config-router)#network 1.1.1.1 255.255.255.0 area 0
Ruijie(config-router)#network 2.2.2.1 255.255.255.0 area 0
Ruijie(config-router)#network 10.10.10.1 255.255.255.255 area 0
Ruijie(config-router)#exit
```

# Configure other nodes.   The configurations of other nodes are the same as LSR A.

- On each node, enable global MPLS forwarding, interface-based MPLS label packet forwarding as well as interface-based LDP.

# Configure LSR A.

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)#mpls ip
Ruijie(config)#interface gigabitEthernet 1/1
Ruijie(config-if-GigabitEthernet 1/1)#label-switching
Ruijie(config-if-GigabitEthernet 1/1)#mpls ip
Ruijie(config-if-GigabitEthernet 1/1)#exit
Ruijie(config)#interface gigabitEthernet 1/2
Ruijie(config-if-GigabitEthernet 1/2)#label-switching
Ruijie(config-if-GigabitEthernet 1/2)#mpls ip
Ruijie(config-if-GigabitEthernet 1/2)#exit
```

# Configure other nodes.   The configurations of other nodes are the same as LSR A.

- Configure LDP to enable MPLS traffic forwarding.

# Configure LSR A.

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)#mpls router ldp
Ruijie(config-mpls-router)#ldp router-id interface loopback 0 force
```

# Configure other nodes. The configurations of other nodes are the same as LSR A.

- Configure per packet scheduling and establish equal cost LSPs for all equal cost routes.

# Configure LSR A.

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)#mpls load-balance packet
Ruijie(config)#mpls router ldp
Ruijie(config-mpls-router)#ldp-ecmp for all-fec
```

Verification

# Display MPLS route information on LSR A.

```
Ruijie#show mpls rib
Status codes: m – main entry, b – backup entry, * - active, s – stale.
Default VRF:
LSP Information   Total
STATIC LSP        0
LDP LSP          2
RSVP LSP         0
BGP LSP          0
L3VPN LSP        0
LDP LSP:
-------------------------------------------------------------------------------------
----------------
FEC           In/Out Label    In/Out IF      Nexthop
* 10.10.10.4/32   -/1024          -/Gi1/1     1.1.1.2
* 10.10.10.4/32   -/2001          -/Gi1/2     2.2.2.2
-------------------------------------------------------------------------------------
----------------
```

# Display the MPLS forwarding table on LSR A.

```
Ruijie# show mpls forwarding-table
Label Operation Code:
PH--PUSH label
PP--POP label
SW--SWAP label
SP—-SWAP topmost label and push new label
DP—-DROP packet
PC—-POP label and continue lookup( IP or Label )
PI--POP label and do ip lookup forward
PN--POP label and forward to nexthop
PM--POP label and do MAC lookup forward
PV--POP label and output to VC attach interface
IP--IP lookup forward
Local  Outgoing  OP   FEC           Outgoing      Next Hop
```

```
label  label                         interface
--      1024     PH   10.10.10.4/32   Gi1/1          1.1.1.2
--      2001     PH   10.10.10.4/32   Gi2/2          2.2.2.2
......
```

Link Layer Protocol Configuration

HDLC Configuration

Understanding HDLC

The RGOS supports the Cisco HDLC private protocol. Unlike the ISO HDLC, the Cisco HDLC protocol uses the SDLC frame format and supports synchronous and full-duplex operation, but does not support the traffic control like the ISO HDLC. It is an unreliable connection. After this protocol is encapsulated, the reliable connection is implemented at the upper layer. The HDLC features high efficiency and simple implementation and is a point-to-point (PTP) link protocol.

Note

Reliable connection refers to the message acknowledgement mechanism that is used during data communication. A lost packet will be retransmitted and the connection is interrupted if a packet times out. The PTP protocol means that communication parties are in one-to-one relationship. PPP and SLIP are also PTP protocols, whereas X.25 and Frame Relay are point-to-multipoint protocols.

The working principle of the HDLC is illustrated in the following phases:

- Negotiation and connection establishment: Two parties of the HDLC link send a link detection negotiation message to each other every 10 seconds. The messages are received or sent based on the sequence numbers of messages. Disorder of sequence numbers results in link disconnection. This kind of message that is used to detect whether a PTP link is active is called the keepalive message.
- Message transmission: The IP messages are encapsulated at the HDLC layer. During data transmission, the keepalive message negotiation is still working to detect whether the link is valid.
- Timeout disconnection: When the interface encapsulated with HDLC cannot receive the acknowledgement from the peer to increment the sequence number for 3 times continuous (or 6 times when the packet receiving speed is over 1000 packets/second), the link state changes from UP to Down. At this time, the link is in down state, and data communication fails.

Configuring HDLC

HDLC Configuration Task List

The HDLC configuration is rather simple and involves only the following tasks.

- Configuring the Interface Encapsulation Protocol
- Configuring the Keepalive Time

Configuring the Interface Encapsulation Protocol

The protocol encapsulated on the synchronous interface is HDLC by default.

Use the following command to change the protocol encapsulated on the interface to HDLC.

Command	**Function**

Command	Function
Ruijie(config-if)#**encapsulation hdlc**	Encapsulates the HDLC protocol.

Configuring the Keepalive Time

For the HDLC encapsulated on a synchronous interface, only two parameters are configurable: interval at which the keepalive message is sent and the maximum timeout time of the keepalive message. The interval is 10 seconds by default.

Use the following command to set the interval and maximum timeout time of the keepalive message based on the link traffic.

Command	Function
Ruijie(config-if)#**keepalive** *second*s	Sets the interval at which the keepalive message is sent and the maximum timeout time of the keepalive message. The maximum timeout time is optional. The range of the interval is from 1 to 32767. The range of the maximum timeout time is from 1 to 255.

Monitoring and Maintaining HDLC

Command	Function
Ruijie#**debug hdlc events**	Turns on the HDLC link status event debugging switch.
Ruijie#**debug hdlc packets**	Turns on the HDLC message receiving/transmitting debugging switch.

1) debug hdlc events

If an interface (for example, Serial1/0) is encapsulated with HDLC, the following information is printed during the keepalive message negotiation:

```
%Interface serial 1/0 : receive one HDLC keepalive packet.
%Interface serial 1/0 send one keepalive packet:
   my_seq = 21, my_seen = 20, your_seen = 16
   line protocol is UP, not in loopback state.
%Interface serial 1/0 : receive one HDLC keepalive packet.
%Interface serial 1/0 send one keepalive packet:
   my_seq = 22, my_seen = 21, your_seen = 17
   line protocol is UP, not in loopback state.
```

Where, my_seq is the sequence number of the message sent by the local router, my_seen is the sequence number of the HDLC keepalive message recognized by the peer router, and your_seen means the sequence number of the peer router recognized by the local router. The sequence numbers are incremental. See the following debug information:

```
%Interface serial 1/0 : receive one HDLC keepalive packet.
%Interface serial 1/0 send one keepalive packet:
   my_seq = 21, my_seen = 20, your_seen = 16
   line protocol is UP, not in loopback state.
```

```
%Interface serial 1/0 send one keepalive packet:
   my_seq = 22, my_seen = 20, your_seen = 16
   line protocol is UP, not in loopback state.
%Interface serial 1/0 send one keepalive packet:
   my_seq = 23, my_seen = 20, your_seen = 16
   line protocol is UP, not in loopback state.
```

The local sequence number my_seq increments according to the keepalive time, but the keepalive message of the peer router is not received. The my_seen is always 20. The local party has no way to know the acknowledgement for the increment of your_seen. This means that the message of the peer router cannot reach the local HDLC protocol layer during communication possibly because the peer router is shut down or a fault occurs during line transmission.

2) debug hdlc packets

Use this command to turn on the HDLC receiving/transmitting message debug switch to print the messages received or to be sent by the HDLC, including the message length and received message type. If the message is longer than 64 bytes, only the first 64 bytes are printed, as shown below:

```
Interface serial 1/0 HDLC input:
  packet->len = 22(0x16):
  8F 00 80 35 00 00 00 02 00 00 00 16 00 00 00 1A
  FF FF 00 5C E2 53
  packet->pkt_type = 3(PDD_RARP)
Interface serial 1/0 HDLC output:
  packet->len = 22(0x16):
  8F 00 80 35 00 00 00 02 00 00 00 1B 00 00 00 16
  FF FF 00 5F 8E D9
```

PPP and MP Protocol Configuration

PPP and MP Protocol Introduction

The PPP (Point-to-Point Protocol) is a kind of link layer protocol providing bearer for network data packets over point-to-point links. PPP defines a whole set of protocols, including LCP (Link Control Protocol), NCP (Network Control Protocol) and authentication protocols (PAP and CHAP). PPP is widely used for its ability of authentication, easy to expand and support of synchronization and asynchronization. For the PPP specifications, see RFC 1661.

PPP Working Process and Principle

- The PPP performs LCP negotiation before line setup, including operation mode (SP or MP), authentication method, and maximum transmission unit.
- The PPP enters the Line Establish phase after LCP negotiation. At this time, the LCP state is Opened, indicating that the link has been established.
- If authentication (the remote end authenticates the local end or vice versa) is enabled in the configuration, the PPP enters the Authenticate phase to start CHAP or PAP authentication.
- If authentication fails, it enters the Terminate phase to remove the link, and the LCP status turns to Closed; if the authentication succeeds, it enters the Network consultation phase (NCP), here, the LCP status is still Opened but that of IPCP and IPXCP turns from Closed to Opened.
- The support of NCP negotiation includes IPCP, IPXCP and BRIDGECP negotiation. IPCP negotiation mainly includes the IP addresses of both parties; IPXCP negotiation mainly includes network IDs and node numbers; BRIDGECP negotiation mainly includes the MAC addresses of both parties, MAC address type, spanning tree and Bridge ID. One or more network layer protocols are selected and configured through NCP negotiation. After the successful configuration of each selected network layer protocol, this network layer protocol can send messages through this link.
- This link will keep available for communication, until a definite LCP or NCP frame closes it, or some external events happen.

PPP Authentication Mode

The PPP supports two kinds of authentication modes: PAP and CHAP.

1. The PAP is a two-handshake authentication, and the password is in clear text.The PAP authentication process is as follows:
   1) The party to be authenticated sends the username and password to the authenticating party.
   2) The authenticating party checks whether this user name exists and whether the password is correct according to the user configuration, and then returns a response accordingly.

2. The CHAP refers to challenge handshake authentication protocol, and the password is in cipher text (key). The process of CHAP authentication is as follows:
    1) The authenticating party sends some random reports to the party to be authenticated.
    2) The authenticated party encrypts the random messages using its own password and MD5 algorithm, and sends the generated cipher text back to the authenticating party.
    3) The authenticating party encrypts the original random reports with the stored password of the authenticated party and the MD5 algorithm, compares the two cipher texts, and then returns the relevant response according to the comparison results.

MP Protocol Introduction

The Multi-Link PPP (MP) binds the PPP of multiple physical links to a single logical interface, aiming to increase link bandwidth. Any physical link that supports PPP can enable MP and be bound to the same logical interface Dialer port. The MP allows fragment of the messages on the network layers like IP. The fragments of the message are transmitted via multiple links and arrive at the same destination at the same time, resulting in summarization of the bandwidth of all links.

The operational process of the MP is as follows:

After the negotiation of general LCP parameters is completed, the PPP initiates the MP request again. If the peer link supports MP and responds properly, it is bound to the logical interface together with the other physical links for further NCP (such as IPCP) negotiation. If negotiation succeeds, all MP physical links use the network address of the same logical interface.

PPP Configuration

PPP Configuration Task List

This chapter describes how to configure PPP in the dedicated line mode (including synchronous interface and asynchronous interface). For the PPP configuration for the dialup connection available in the serial interface, additional configurations are needed in addition to the following ones. See Dialup Configuration Guide for details.

- Configure the interface encapsulation protocol
- Configure the PPP CHAP authenticated party
- Configure the PPP CHAP authenticating party
- Configure the PPP PAP authenticated party
- Configure the PPP PAP authenticating party
- Configure the PPP compression mode

Configuring the Encapsulation Protocol on the Interface

To configure the PPP, encapsulate the PPP on the interface. To encapsulate the PPP, run the following commands in the interface configuration mode:

Command	Function
Ruijie(config-if)#**encapsulation ppp**	Encapsulate the PPP on the interface.
Ruijie(config-if)#**no encapsulation ppp**	Remove the PPP encapsulation on the interface.

Configuring the PPP CHAP Authenticated Party

The CHAP authentication generally involves authenticating party and authenticated party. The CHAP negotiation is initiated by the authenticating party, and the authenticated party sends only the username and password for the use of the PPP authentication. By default, the authenticated party sends its own hostname as the PPP username.

To configure the PPP CHAP authenticated party, use the following command at interface configuration mode:

Command	Function
Ruijie(config-if)# **ppp chap hostname** *hostnmae*	Specify the hostname for PPP CHAP authentication.
Ruijie(config-if)# **ppp chap password** {**0|7**} *password*	Specify the password for PPP CHAP. authentication

If the hostname used by the authenticating party is known, the following configuration can be used:

Command	Function
Ruijie(config-if)# **ppp chap hostname** *hostnmae*	Specify the hostname for PPP CHAP authentication.
Ruijie(config)# **username** *username* password {0|7} *password*	Create user database records for the authenticating party hostname, with passwords consistent at both ends.

Note

There are clear text password and cipher text password, "0" for clear text password and 1-7 for cipher text password. The default input method is the clear text password. In this manual, all places involving password setting is suitable for the above rule. In the interconnection with other manufacturers, only the clear text password is accepted. For the configuration of bidirectional authentication, the configuration of authenticating party is also needed.

Configuring the PPP CHAP Authenticating Party

The PPP CHAP authenticating party initiates the authentication proactively. Since the username and password from the peer router shall be validated, the authenticating party shall create and maintain a local user database. To configure the PPP CHAP authenticating party, use the following command at interface configuration mode:

Command	Function
Ruijie(config-if)# **ppp authentication chap [callin]**	Enable the PPP authentication and specify the PPP CHAP authentication method.
Ruijie(config-if)# **no ppp authentication chap**	Disable the PPP CHAP authentication.
Ruijie(config)# **username** *username* **password {0|7}** *password*	Create the user database record.

The authenticating party has the username and related password configured in the user database, where the username is the PPP hostname of the peer router (the authenticated party).

Note

The Callin is an optional command option. With it configured, the CHAP authentication is not initiated unless the peer router (authenticated party) dials to connect the network in dialup mode. For the PPP connection that is set up because the local router dials out, the CHAP authentication is not initiated. Therefore, this command does not affect the dedicated line PPP negotiation.

Configuring the PPP PAP Authenticated Party

The PAP authentication involves authenticating party and authenticated party. For the setting of PAP authenticated party for PPP, run the following commands:

Command	**Function**
Ruijie(config-if)# **ppp pap sent-username** *username* **password {0|7}** *password*	Specify the username and password for PPP PAP authentication.
Ruijie(config-if)# **no ppp pap sent-username**	Remove the PPP PAP authentication settings.

Configuring the PPP PAP Authenticating Party

The commands for setting the PPP PAP authenticating party are as follows:

Command	**Function**
Ruijie(config-if)# **ppp authentication pap [callin]**	Configure the PPP PAP authenticating party
Ruijie(config)# **username** *username* **password {0|7}** *password*	Create the user database record.

Note

The Callin is an optional command option. With it configured, the PAP authentication is not initiated unless the peer router (authenticated party) dials to connect the network via dialup mode. For the PPP connection that is set up because the local router dials out, the PAP authentication is not initiated. Therefore, this command does not affect the dedicated line PPP negotiation.

Configure the PPP Negotiation Parameters

During the PPP negotiation, both LCP and IPCP have timeout periods. Once the period expires, the LCP resends requests. This period can be set by using this command to coordinate the negotiation time in the interconnection with heterogeneous devices.

Command	**Function**
Ruijie(config-if)#**ppp negotiation-timeout** *seconds*	Configure the PPP LCP negotiation time.
Ruijie(config-if)#**no ppp negotiation-timeout**	Restore the PPP negotiation time to the default.

Configuring Sender of PPP LCP Extended Configuration Option

The sender of the PPP LCP configuration option actively adds the extended configuration option to LCP configuration request packets. The IMSI number, the serial number (SN) of the local router, or the MAC address of the local router is sent to the peer end by being carried in the extended configuration option. Run the following command in interface configuration mode to configure the sender of the PPP LCP extended configuration option:

Command	Function
Ruijie(config-if)#**ppp lcp send-option** {**imsi|serial-number|mac-address**}	Sends the LCP extended configuration option carrying the IMSI number, the SN, or the MAC address.

Note

Run the **ppp lcp send-option imsi** command to send the LCP extended configuration option carrying the IMSI number. Run the **ppp lcp send-option serial-number** command to send the LCP extended configuration option carrying the SN. Run the **ppp lcp send-option mac-address** command to send the LCP extended configuration option carrying the MAC address.

Configuring Receiver of PPP LCP Extended Configuration Option

The receiver of the PPP LCP extended configuration option needs to identify the PPP LCP extended configuration option sent from the remote router and send the information in the extended configuration option to the AAA server. Run the following command in interface configuration mode to configure the receiver of the PPP LCP extended configuration option:

Command	Function
Ruijie(config-if)#**ppp lcp accept-option**	Identifies the PPP LCP extended configuration option and sends the information in the extended configuration option to the AAA server.

Note

Run the **ppp lcp accept-option** command to identify the PPP LCP extended configuration option carrying IMSI number. Run the **ppp lcp accept-option** command to identify the PPP LCP extended configuration option carrying the SN. Run the **ppp lcp accpet-option** command to identify the PPP LCP extended configuration option carrying the MAC address.

The preceding command can also send the information in the extended configuration option to the AAA server if the AAA function is enabled.

In addition to the **ppp lcp accept-option** command, run the **force-local-lcp** command on the LNS device to transmit the IMSI, the SN, or the MAC address via the PPP LCP extended configuration option. In addition, the functions must be normal to ensure that the IMSI/MAC/SN extended configuration option is identified.

The LAC device of the operator may not support LCP re-negotiation and the configuration of the **force-local-lcp** command will result in a PPP negotiation failure. As a result, the IMSI/MAC/SN authentication function is unavailable.

Configuring Function of Accepting/Rejecting/Ignoring Received ACFC Negotiation Requests

During PPP link negotiation, run the following command in interface configuration mode to determine whether to accept the Address-and-Control-Field-Compression (ACFC) negotiation request sent from the peer end:

Command	Function
Ruijie(config-if)#**ppp acfc remote** {**apply**|**reject**|**ignore**}	Configures how the local router processes the PPP configuration request packet that carries the ACFC option and that is sent from the remote device.

Note

Run the **ppp acfc remote apply** command to accept ACFC negotiation from the peer end. That is, ACFC is enabled for PPP frames from the peer end, but is enabled or disabled for PPP frames from the local end based on the interface type.

Run the **ppp acfc remote reject** command to reject ACFC negotiation from the peer end.

Run the **ppp acfc remote ignore** command to ignore ACFC negotiation from the peer end . That is, ACFC can be enabled or disabled for PPP frames from the peer end, but is disabled for PPP frames from the local end.

If the **ppp acfc remote** command conflicts with the **ppp acfc local** command, the later configured command shall prevail.

Configuring Function of Initiating/Forbidding ACFC Option

During PPP link negotiation, run the following command in interface configuration mode to determine whether to initiate ACFC negotiation from the local end:

Command	Function
Ruijie(config-if)#**ppp acfc local** {**request**|**forbid**}	Configures how the local router processes the ACFC option in the to-be-sent PPP configuration request.

Note

Run the **ppp acfc local request** command to initiate ACFC negotiation on the local end, that is, the PPP LCP configuration request packet to be sent from the local end carries the ACFC option.

Run the **ppp acfc local forbid** command to forbid ACFC negotiation on the local end, that is, the PPP LCP configuration request packet to be sent from the local end does not carry the ACFC option and the configuration request packet that carries the ACFC option and that is sent from the peer end is rejected.

If the **ppp acfc local** command conflicts with the **ppp acfc remote** command, the later configured command shall prevail.

Configuring Function of Accepting/Rejecting/Ignoring Received PFC Negotiation Requests

During PPP link negotiation, run the following command in interface configuration mode to determine whether to accept the protocol field compression (PFC) negotiation request sent from the peer end:

Command	Function
Ruijie(config-if)#**ppp pfc remote** {**apply**|**reject**|**ignore**}	Configures how the local router processes the PPP configuration request packet that carries the PFC option and that is sent from the remote device.

Note

Run the **ppp pfc remote apply** command to accept PFC negotiation from the peer end. That is, PFC is enabled for PPP frames form the peer end, but is enabled or disabled for PPP frames from the local end based on the interface type.

Run the **ppp pfc remote reject** command to reject PFC negotiation from the peer end.

Run the **ppp pfc remote ignore** command to ignore PFC negotiation from the peer end. That is, PFC can be enabled or disabled for PPP frames from the peer end, but is disabled for PPP frames from the local end.

If the **ppp pfc remote** command conflicts with the **ppp pfc local** command, the last configured command shall prevail.

Configuring Function of Initiating/Forbidding PFC Option

During PPP link negotiation, run the following command in interface configuration mode to determine whether to initiate PFC negotiation from the local end:

Command	Function
Ruijie(config-if)#**ppp pfc local** {**request**|**forbid**}	Configures how the local router processes the PFC option in the to-be-sent PPP configuration request.

Note

Run the **ppp pfc local request** command to initiate PFC negotiation on the local end, that is, the PPP LCP configuration request packet to be sent from the local end carries the PFC option.

Run the **ppp pfc local forbid** command to forbid PFC negotiation on the local end, that is, the PPP LCP configuration request packet to be sent from the local end does not carry the PFC option and the configuration request packet that carries the PFC option and that is sent from the peer end is rejected.

If the **ppp pfc local** command conflicts with the **ppp pfc remote** command, the last configured command shall prevail.

MP configuration

Configuring MP on the Dialer Interface

MP Configuration Task List

The MP can be implemented by configuring the rotary-group on the physical interface layer and binding the dialer logical interface. This chapter describes only the multilink PPP binding the dialer interface and the synchronous serial interface. For the configuration of the dialup multilink of asynchronous serial interface, see DDR Configuration Guide. The list of configuration tasks for the multilink of Dialer interface binding synchronous serial interface is as follows:

- Configuring the synchronous serial interface
- Encapsulating PPP link protocol
- Configuring dialer in-band
- Configuring rotary-group
- The configuration of the rotary-group needs the DDR. So, if the **dialer in-band** has not been configured in advance, it will be automatically configured in configuring **dialer rotary-group**.
- Creating the logical interface dialer
- When you create the logical interface dialer, the physical interface with the rotary group configured must be existent, which maintains the binding by consistency of the rotary group number and the logical interface dialer number. If the logical interface is to be deleted, it is required to delete the rotary group on the physical interface first.
- Configuring the ppp multilink
- Configuring the dialup filtering rule

Configuring the synchronous serial interface

To configure the synchronous serial interface for multilink binding:

Command	Function
Ruijie(config)# **interface serial** *interface-number*	Enter the configuration mode of the specified serial interface.

Encapsulating PPP link protocol

The multilink PPP is a PPP at first. So, it is required to encapsulate the PPP link protocol first no matter whether it is on a physical interface or a logical interface.

Command	Function
Ruijie(config-if)#**encapsulation ppp**	Encapsulate the PPP link protocol.
Ruijie(config-if)#**no encapsulation ppp**	Cancel the encapsulation of the PPP link protocol.

Configuring dialer in-band

To configure the multilink, it is required to configure the DDR on the serial interface configuration layer at first by using the following command, which is the prerequisite for configuring rotary-group:

Command	Function
Ruijie(config-if)#**dialer in-band**	Set the DDR configuration.
Ruijie(config-if)#**no dialer in-band**	Cancel the DDR configuration.

Configuring rotary-group

This command binds the physical interface to the rotary group of logical interface to enable the multilink binding.

Command	Function
Ruijie(config-if)#**dialer rotary-group** *group-number*	Set the number of the rotary group.
Ruijie(config-if)#**no dialer rotary-group** *group-number*	Delete the rotary group.

Note

The configuration of the rotary-group needs the DDR. So, if the **dialer in-band** has not been configured in advance, it will be automatically configured in configuring **dialer rotary-group**.

Creating the logical interface dialer

When the rotary group is set, it is necessary to create the logical interface dialer that individual physical interfaces are bound to. The interface number must be the same as the number of the rotary group. After the logical interface dialer is created, it enters into the dialer logical interface configuration layer. Accordingly, it is required to encapsulate the PPP link protocol, configure the PPP multilink, configure the dialup filtering rule, and so on.

Command	Function
Ruijie(config)#**interface dialer** *group-number*	Create the logical interface dialer.
Ruijie(config)#**no interface dialer** *group-number*	Delete the logical interface dialer.

Note

When you create the logical interface dialer, the physical interface with the rotary group configured must be existent, which maintains the binding by consistency of the rotary group number and the logical interface dialer number. If the logical interface is to be deleted, it is required to delete the rotary group on the physical interface first.

Configuring the ppp multilink

To set the multilink PPP, it is required to configure the **PPP multilink** command on the logical interface to specify the logical interface to use the multilink negotiation mode.

Command	Function
Ruijie(config-if)#**ppp multilink**	Set the multilink negotiation mode.
Ruijie(config-if)#**no ppp multilink**	Cancel the setting of the PPP multilink negotiation mode.

Configuring the dialup filtering rule

On the logical interface, it is required to configure the dialup filtering rule. This rule is powerful enough to use with the ACL. Here is the simplified description for the functions of the **dialer-list**.

Command	Function
Ruijie(config-if)#**dialer-group** *group-number*	Set the dialer group.
Ruijie(config)#**dialer-list** *group-number* **protocol ip permit**	Set the common dialer group list allowing the access to IP packets.

Configuring MP on the Multilink Interface

MP Configuration Task List

The MP can be implemented by configuring the multilink group on the physical interface layer and binding the multilink logical interface. This chapter describes only the multilink PPP of the multilink interface binding the synchronous serial interface. The list of configuration tasks of multilink interface binding synchronous serial interface is as follows:

- Creating the logical interface multilink
- Configuring the synchronous serial interface
- Encapsulating PPP link protocol
- Configuring the ppp multilink
- Configuring the ppp multilink group

Creating the logical interface multilink

To configure the MP binding of the multilink interface, create the logical interface multilink binding individual physical interfaces. After the logical interface multilink is created, it enters into the multilink logical interface configuration layer, where the PPP encapsulation and MP are enabled by default.

Command	Function
Ruijie(config)#**interface multilink** *group-number*	Create the logical interface multilink.
Ruijie(config)#**no interface multilink** *group-number*	Delete the logical interface multilink.

Caution Up to 72 multilink interfaces are supported.

Configuring the synchronous serial interface

To configure the synchronous serial interface for multilink binding, execute the following command:

Command	Function
Ruijie(config)# **interface serial** *interface-number*	Enter the specified serial interface configuration mode.

Encapsulating PPP link protocol

The multilink PPP is a PPP at first. So, it is required to encapsulate the PPP link protocol first on the synchronous serial interface.

Command	Function
Ruijie(config-if)#**encapsulation ppp**	Encapsulate the PPP link protocol.
Ruijie(config-if)#**no encapsulation ppp**	Remove the encapsulation of PPP link protocol.

Configuring the ppp multilink

To set the synchronous interface to use the multilink negotiation mode, execute the **PPP multilink** command on the synchronous serial interface.

Command	Function
Ruijie(config-if)#**ppp multilink**	Set the multilink negotiation mode.
Ruijie(config-if)#**no ppp multilink**	Remove the setting of the PPP multilink negotiation mode.

Configuring the ppp multilink group

To bind the synchronous serial interface to the group of logical interface for multilink binding, execute the following command.

Command	Function
Ruijie(config-if)# **ppp multilink group** *group-number*	Set the number of the multilink group.
Ruijie(config-if)#**no ppp multilink group** *group-number*	Remove the number of the multilink group.

Note The parameter group-number configured here is the same as the multilink interface number in the creation of the multilink. When configuring the ppp multilink group, the multilink interface must exist accordingly. The physical interface uses the multilink group number and the logical interface number of multilink interface to maintain binding. If the logical interface is to be deleted, it is required to delete the multilink group on the physical interface first.

Configuring MP on the Virtual-Template Interface

MP Configuration Task List

The MP can be implemented by configuring the MP on the physical interface layer and binding the virtual template interface. This chapter describes only the multilink PPP of the virtual template interface binding the synchronous serial interface. For details of the virtual template, see VPN Configuration Guide. The list of configuration tasks for the virtual template interface binding synchronous serial interface is as follows:

- Creating the virtual template interface
- Creating the bounded virtual template interface
- For the Virtual-Template multilink binding, in the setup of the binding, a virtual access interface Virtual-Access is created automatically as the binding interface. This command specifies which virtual template copy configuration acts as the settings of the virtual access interface during the creation of the virtual access interface.
- Configuring the synchronous serial interface
- Encapsulating the PPP link protocol
- Configuring the ppp multilink

Creating the virtual template interface

To configure the MP in the Virtual-Template method, it is required to create the Virtual-Template interface. After the Virtual-Template interface is created, it enters into the virtual template interface configuration layer. Accordingly, it is required to encapsulate the PPP link protocol, configure the PPP multilink, configure the dialup filtering rule, and so on.

Command	Function
Ruijie(config)#**interface virtual-template** *number*	Create/configure the specified virtual -template interface
Ruijie(config)#**no interface virtual-template** *number*	Delete the specified virtual -template interface

Creating the bounded virtual template interface

After the virtual template interface is created, the **multilink virtual-template** command is used to specify which virtual template copy binds the interface parameters.

Command	Purpose
Ruijie(config)# **multilink virtual-template** *number*	Specify the virtual template for the MP bundle interface to clone interface parameters
Ruijie(config)# **no multilink virtual-template** *number*	Delete the virtual template for the MP bundle interface to clone interface parameters

Note For the Virtual-Template multilink binding, in the setup of the binding, a virtual access interface Virtual-Access is created automatically as the binding interface. This command specifies which virtual template copy configuration acts as the settings of the virtual access interface during the creation of the virtual access interface.

Configuring the synchronous serial interface

To configure the synchronous serial interface for multilink binding:

Command	Function
Ruijie(config)# **interface serial** *interface-number*	Enter the configuration mode of the specified serial interface

Encapsulating the PPP link protocol

The multilink PPP is a PPP at first. So, it is required to encapsulate the PPP link protocol first no matter whether it is on a synchronous serial interface or a virtual template interface.

Command	Function
Ruijie(config-if)#**encapsulation ppp**	Encapsulate PPP link protocol
Ruijie(config-if)#**no encapsulation ppp**	Cancel the encapsulation of PPP link protocol

Configuring the ppp multilink

To set the virtual template interface and physical interface to use the multilink negotiation mode, execute this command on the virtual template interface and synchronous interface.

Command	Function
Ruijie(config-if)#**ppp multilink**	Set the multilink negotiation mode.
Ruijie(config-if)#**no ppp multilink**	Remove the setting of the PPP multilink negotiation mode.

Note It is not necessary to configure multilink group or rotary-group on the physical interface for virtual template-based binding. It only the **ppp multilink** command is configured on the physical interface but no configuration for which link group or dialup rotary group to belong to, the physical interface will belong to the multilink binding in virtual template mode.

PPP Monitoring and Maintenance

Showing the Protocol Interface Information

Run the following command to show the PPP protocol interface information, which is the first step to debug PPP:

Command	**Function**
Ruijie#**show interface serial** *interface-number*	Show the information of the serial interface.

Take Serial1/0 as an example. The following information in printed after the command is entered:

```
serial 1/0 is UP  , line protocol is UP
Hardware is Infineon DSCC4 PEB20534 H-10 serial
Interface address is: 100.100.100.1/24
 MTU 1500 bytes, BW 2000 Kbit
 Encapsulation protocol is PPP, loopback not set
 Keepalive interval is 10 sec , set
 Carrier delay is 2 sec
 RXload is 1 ,Txload is 1
 LCP Open
 Open: ipcp
 Queueing strategy: WFQ
 5 minutes input rate 30 bits/sec, 0 packets/sec
 5 minutes output rate 30 bits/sec, 0 packets/sec
 49 packets input, 786 bytes, 0 no buffer
 Received 1 broadcasts, 0 runts, 0 giants
 0 input errors, 0 CRC, 0 frame, 0 overrun, 0 abort
 47 packets output, 768 bytes, 0 underruns
 0 output errors, 0 collisions, 1 interface resets
 1 carrier transitions
  V35 DTE cable
 DCD=up  DSR=up  DTR=up  RTS=up  CTS=up
```

First check the link status from the physical layer. The five signals (DCD, DSR, DTR, RTS, CTS) in the last row determine whether the serial is UP or not. This is the prerequisite for PPP Line Protocol UP.

Then, check whether the PPP negotiation is successful by checking whether the LCP status is UP and whether the IPCP status is UP. If yes for both, the line protocol is in the up status, and the link layer shall be available for communication.

Last, refer to the data receiving/transmitting conditions at the bottom layer. Packet Input and Output indicate the number of messages received and transmitted. If there is no interface reset, the message transmission is successful. If there is no number of drops in the Input Queue, it means all messages are received successfully.

PPP Debugging Information

In case of problem with the PPP link layer negotiation, run the following command to debug the PPP:

Command	**Function**
Ruijie# **debug ppp packet**	Print message debug information during PPP

Command	Function
	communication
Ruijie# **debug ppp negotiation**	Print negotiation debug information during PPP communication
Ruijie# **debug ppp authentication**	Print authorization debug information during PPP communication

Packet debugging

In the privileged command layer, enter the **PPP Packet** debug command:

```
Ruijie#debug ppp packet
PPP: serial 1/0 [S] LCP CONFREQ id 3 len 10
  MAGICNUMBER (6) 0x0 0x2b 0x39 0x1b
%LINK CHANGED: Interface serial 1/0, changed state to up
PPP: serial 1/0 [R] LCP CONFREQ id 6 len 10
  MAGICNUMBER (6) 0x29 0xbd 0xea 0xeb
PPP: serial 1/0 [S] LCP CONFACK id 6 len 10
  MAGICNUMBER (6) 0x29 0xbd 0xea 0xeb
PPP: serial 1/0 [R] LCP CONFACK id 3 len 10
  MAGICNUMBER (6) 0x0 0x2b 0x39 0x1b
PPP: serial 1/0 LCP up
PPP: serial 1/0 PPP up.
PPP: serial 1/0 [S] IPCP CONFREQ(2) id 10 len 2
  Address (6) 0x64 0x64 0x64 0x1
PPP: serial 1/0 [R] IPCP CONFREQ(3) id 10 len 2
  Address (6) 0x64 0x64 0x64 0x2
PPP: serial 1/0 [S] IPCP CONFACK(3) id 10 len 2
  Address (6) 0x64 0x64 0x64 0x2
PPP: serial 1/0 [S] LCP PROTREJ id 4 len 10 protocol = 0x82070103
PPP: serial 1/0 [R] IPCP CONFACK(2) id 10 len 2
  Address (6) 0x64 0x64 0x64 0x1
%LINE PROTOCOL CHANGE: Interface serial 1/0, changed state to UP
Ruijie#
PPP: serial 1/0 [S] LCP ECHOREQ id 1 len 12 magic 0x2b391b
```

The above debugging information is for all messages from the start of the PPP negotiation to the Line Protocol Up, without authentication. Pay attention to the bolded debug information: Both routers send the LCP CONFREQ to each other and then respond with the LCP CONFACK. Then it enters into the IPCP negotiation. Also pay attention to the bolded contents. Both routers send IPCP CONFREQ to each other and attach their own IP addresses, and then send the IPCP CONFACK response with the peer IP address attached after the IPCP CONFREQ request is received. Now the PPP negotiation of the interface succeeds.

PPP negotiation debugging information

The negotiation debugging information is used for the debugging trace purpose in negotiating the PPP parameters. For example, in the CHAP authentication of PPP, the **debug ppp negotiation** command can trace the following parameter negotiation information:

```
Ruijie#
PPP: serial 1/0 reset lcp options
```

```
PPP: serial 1/0 sending OPCODE_CONFREQ, type = 5 (LCP_MAGICNUMBER), value = 0x10a5df
%LINK CHANGED: Interface serial 1/0, changed state to up
LCP: received config , type = 5 (MAGICNUMBER) value = 0x29cbca60 acked
PPP: serial 1/0 OPCODE_CONFACK received, type = 5 (LCP_MAGICNUMBER), value = 0x10a5df
PPP: serial 1/0 state = Acksent ppp_recv_confack(0xc021): rcvd id 5
PPP: serial 1/0 reset ipcp options
IPCP: serial 1/0 sending OPCODE_CONFREQ, type = 3 (IPCP_ADDRESS), Address = 100.100.100.1
IPCP: serial 1/0 received ADDR : her address 100.100.100.2 (ACK)
IPCP: ipcp_do_req_cb: returning OPCODE_CONFACK.
IPCP: serial 1/0 OPCODE_CONFACK received, type = 3 (IPCP_ADDRESS), Address = 100.100.100.1
PPP: serial 1/0 state = Acksent ppp_recv_confack(0x8021): rcvd id 3
IPCP: serial 1/0 install route to 100.100.100.2
%LINE PROTOCOL CHANGE: Interface serial 1/0, changed state to UP
```

Pay attention to the bolded contents, which are the options for the LCP and IPCP negotiation parameters.

PPP authentication debugging information

The debug of the authentication information is used to trace the PPP authentication. With PAP as an example, the PAP authenticating party prints the following debugging information:

```
PPP: serial 1/0 PAP authenticating peer DRO36
PPP: serial 1/0 Remote passed PAP authentication sending Auth-Ack to peer.
PPP: serial 1/0 lcp authentication OK#
```

The CHAP authentication party prints the following information when the sent CHAP password is wrong:

```
PPP: serial 1/0 recv CHAP challenge from Router
PPP: serial 1/0 remote router failed CHAP authentication
PPP: serial 1/0 Remote msg is: Authentication failure
```

Typical PPP Configuration Examples

PPP PAP Authentication Example

The example below shows a PAP configuration, username Ruijie, password Router, authenticating party IP address 1.1.1.1/24, authenticated party IP address 1.1.1.2/24, username and password configured same as the authentication method. Router A is the authenticated party and Router B is the authenticating party.

Router A:

```
Ruijie#config terminal
Ruijie(config)#interface Serial1/0
```

# Configure the IP address

```
Ruijie(config-if)#ip address 1.1.1.2 255.255.255.0
```

# Encapsulate PPP protocol

```
Ruijie(config-if)#encapsulation ppp
```

# Configure the username and password for PAP authentication.

```
Ruijie(config-if)#ppp pap sent-username Ruijie password 0 Router
```

Router B:

```
Ruijie#config terminal
Ruijie(config)#username Ruijie password 0 Router
Ruijie(config)#interface Serial1/0
```

# Configure the IP address

```
Ruijie(config-if)#ip address 1.1.1.1 255.255.255.0
```

# Encapsulate PPP protocol

```
Ruijie(config-if)#encapsulation ppp
```

# Specify the PPP authentication mode

```
Ruijie(config-if)#ppp authentication pap
```

PPP CHAP Authentication Example

The example below shows a PPP CHAP authentication configuration, where the Router A is the authenticating party, IP address is 1.1.1.1/24, hostname is RouterA, password is Router, and the user list contains the hostname RouterB; the Router B is the authenticating party, IP address is 1.1.1.2/24, hostname is RouterB and the password sent is Router.

Router A:

```
Ruijie#config terminal
```

# Set the hostname

```
Ruijie(config)#hostname RouterA
```

# Set the username and password list

```
RouterA(config)#username RouterB password 0 Router
RouterA(config)#username RouterC password 0 Router
RouterA(config)#interface serial1/0
```

# Encapsulate protocol

```
RouterA(config-if)#encap ppp
```

# Set IP address

```
RouterA(config-if)#ip address 1.1.1.1 255.255.255.0
```

# Specify the PPP chap authentication mode

```
RouterA(config-if)#ppp authentication chap
```

Router B:

```
Ruijie#config terminal
```

# Set the hostname

```
Ruijie(config)#hostname RouterB
```

# Use the peer hostname as the username, and the password is the same as that configured on the peer router.

```
RouterB(config)#username RouterA password 0 Router
RouterB(config)#interface serial1/0
```

# Encapsulate protocol

```
RouterB(config-if)#encap ppp
```

# Set IP address

```
RouterB(config-if)#ip address 1.1.1.2 255.255.255.0
```

To keep the administration hostname, the hostname configured for the CHAP authentication can be configured by using the **chap hostname *hostname*** command. In the above example, the Router B as the authenticated party uses the default hostname, and the host for the CHAP authentication is RouterB.

Router B:

```
Ruijie#config terminal
Ruijie(config)#
```

# Use the peer hostname as the username, and the password is the same as that configured on the peer router.

```
R(config)#username RouterA password 0 Router
```

```
Ruijie(config)#interface serial1/0
```

# Encapsulate protocol

```
Ruijie(config-if)#encap ppp
```

# Set IP address

```
Ruijie(config-if)#ip address 1.1.1.2 255.255.255.0
```

# Set the local CHAP authentication hostname

```
Ruijie(config-if)#ppp chap hostname RouterB
```

PPP MS-CHAP Authentication

The following example describes the PPP MS-CHAP authentication configuration. Router A is the authentication party, with the IP address being 1.1.1.1/24, host name being RouterA, and password being Router. The user list of Router A includes the host name of Router B. Router B is the authenticated party, with IP address being 1.1.1.2/24, host name being RouterB, and password being Router.

Router A:

```
Ruijie# config terminal
```

#Set a host name.

```
Ruijie(config)# hostname RouterA
```

#Set a username and password list.

```
RouterA(config)#username RouterB password 0 Router
RouterA(config)#username RouterC password 0 Router
RouterA(config)#interface serial 1/0
```

#Set the encapsulation protocol.

```
RouterA(config-if)# encap ppp
```

#Set an IP address.

```
RouterA(config-if)# ip address 1.1.1.1 255.255.255.0
```

#Set the PPP MS-CHAP authentication mode.

```
RouterA(config-if)# ppp authentication ms-chap
```

Router B:

```
Ruijie# config terminal
```

#Set a host name.

```
Ruijie(config)# hostname RouterB
```

#Set the host name of Router A as the username, and set the password to be the same as that of Router A.

```
RouterB(config)# username RouterA password 0 Router
RouterB(config)# interface serial 1/0
```

#Set the encapsulation protocol.

```
RouterB(config-if)# encap ppp
```

#Set an IP address.

```
RouterB(config-if)# ip address 1.1.1.2 255.255.255.0
```

To reserve the host name, run the **ppp chap hostname** *hostname* command to specify the local host name used for MS-CHAP authentication. In the preceding example, the authenticated party Router B uses the default host name RouterB, which is used for MS-CHAP authentication.

Router B:

```
Ruijie# config terminal
Ruijie(config)#
```

#Set the host name of Router A as the username, and set the password to be the same as that of Router A.

```
Ruijie (config)# username RouterA password 0 Router
Ruijie(config)# interface serial 1/0
```

#Set the encapsulation protocol.

```
            Ruijie(config-if)# encap ppp
```

#Set an IP address.

```
            Ruijie(config-if)# ip address 1.1.1.2 255.255.255.0
```

#Set the host name for local MS-CHAP authentication.

```
            Ruijie(config-if)# ppp chap hostname RouterB
```

PPP MS-CHAP-V2 Authentication

The following example describes the PPP MS-CHAP-V2 authentication configuration. Router A is the authentication party, with the IP address being 1.1.1.1/24, host name being RouterA, and password being Router. The user list of Router A includes the host name of Router B. Router B is the authenticated party, with IP address being 1.1.1.2/24, host name being RouterB, and password being Router.

Router A:

```
            Ruijie# config terminal
```

#Set a host name.

```
            Ruijie(config)# hostname RouterA
```

#Set a username and password list.

```
            RouterA(config)#username RouterB password 0 Router
            RouterA(config)#username RouterC password 0 Router
            RouterA(config)#interface serial 1/0
```

#Set the encapsulation protocol.

```
            RouterA(config-if)# encap ppp
```

#Set an IP address.

```
            RouterA(config-if)# ip address 1.1.1.1 255.255.255.0
```

#Set the PPP MS-CHAP-V2 authentication mode.

```
            RouterA(config-if)# ppp authentication ms-chap-v2
```

Router B:

```
            Ruijie# config terminal
```

#Set a host name.

```
            Ruijie(config)# hostname RouterB
```

#Set the host name of Router A as the username, and set the password to be the same as that of Router A.

```
            RouterB(config)# username RouterA password 0 Router
            RouterB(config)# interface serial 1/0
```

#Set the encapsulation protocol.

```
            RouterB(config-if)# encap ppp
```

#Set an IP address.

```
            RouterB(config-if)# ip address 1.1.1.2 255.255.255.0
```

To reserve the host name, run the **ppp chap hostname** *hostname* command to specify the local host name used for MS-CHAP-V2 authentication. In the preceding example, the authenticated party Router B uses the default host name RouterB, which is used for MS-CHAP-V2 authentication.

Router B:

```
            Ruijie# config terminal
            Ruijie(config)#
```

#Set the host name of Router A as the username, and set the password to be the same as that of Router A.

```
            Ruijie (config)# username RouterA password 0 Router
            Ruijie(config)# interface serial 1/0
```

#Set the encapsulation protocol.

```
Ruijie(config-if)# encap ppp
```

#Set an IP address.

```
Ruijie(config-if)# ip address 1.1.1.2 255.255.255.0
```

#Set the host name for local MS-CHAP-V2 authentication.

```
Ruijie(config-if)# ppp chap hostname RouterB
```

MP Configuration Example

The example below shows a multilink configuration, where two synchronous interfaces serial1/0 and serial1/1 are bound to the logical interface dialer 1 to implement the Multilink PPP.

# Create the loopback interface and use its IP address as the IP address of the dialer logical interface

```
interface Loopback0
ip address 192.168.20.2 255.255.255.0
```

# Configure DDR on the physical interfaces serial1/0 and serial1/1, and specify the rotary group number (1 in this example)

```
interface Serial1/0
no ip address
encapsulation ppp
dialer in-band
dialer rotary-group 1
#
interface Serial1/1
no ip address
encapsulation ppp
dialer in-band
dialer rotary-group 1
```

# Create the dialer 1 logical interface and borrow the IP address of loopback0, set the dialer group, and specify the multilink ppp negotiation method.

```
interface Dialer1
ip unnumbered Loopback0
encapsulation ppp
dialer-group 1
ppp multilink
#
dialer-list 1 protocol ip permit
```

The example below uses two synchronous interfaces serial1/0 and serial1/1 are bound to the logical interface multilink 1 to implement the Multilink PPP.

# Create the loopback interface and use its IP address as the IP address of the multilink logical interface

```
interface Loopback0
  ip address 192.168.20.1 255.255.255.0
```

# Configure ppp multilink on the physical interfaces serial1/0 and serial1/1, and specify the multilink group number 1.

```
interface Serial 1/0
  no ip address
  encapsulation ppp
  ppp multilink
  multilink-group 1
interface Serial 1/1
  no ip address
  encapsulation ppp
  ppp multilink
  multilink-group 1
```

# Create the multilink 1 logical interface and borrow the IP address of loopback0.

```
interface multilink 1
  ip unnumbered Loopback0
  encapsulation ppp
  ppp multilink
```

The example below uses two synchronous interfaces serial1/0 and serial1/1 are bound to the virtual template interface to implement the Multilink PPP.

# Create the loopback interface and use its IP address as the IP address of the virtual template interface

```
interface Loopback0
  ip address 192.168.20.1 255.255.255.0
```

# Configure the PPP multilink on physical interfaces serial1/0 and serial1/1

```
interface Serial 1/0
  no ip address
  encapsulation ppp
  ppp multilink
interface Serial 1/1
  no ip address
  encapsulation ppp
  ppp multilink
```

# Create the virtual-template 1 interface and borrow the IP address of loopback0.

```
interface virtual-template 1
  ip unnumbered Loopback0
  encapsulation ppp
  ppp multilink
multilink virtual-template 1
```

# Specify copying parameters from virtual-template 1, to establish the binding from virtual-access interface to synchronous interface.

Configuration of Sender/Receiver of LCP Extended Configuration Option

In the following example, Router A is configured as the sender of the PPP LCP extended configuration option, so as to send the SN of router A to Router B through the LCP extended configuration option; Router B is configured as the receiver of the PPP LCP extended configuration option, identifies the LCP extended configuration option that carries the SN and that is sent from Router A, and then sends the information in the extended configuration option to the AAA server for authentication.

Router A:

```
Ruijie# config terminal
```

#Set a host name.

```
Ruijie(config)# hostname RouterA
```

#Set the encapsulation protocol.

```
RouterA(config-if)# encap ppp
```

#Set an IP address.

```
RouterA(config-if)# ip address 1.1.1.1 255.255.255.0
```

#Set the sender of the PPP LCP extended configuration option.

```
RouterA(config-if)#ppp lcp send-option serial-number
```

Router B:

```
Ruijie# config terminal
```

#Set a host name.

```
Ruijie(config)# hostname RouterB
```

#Set the encapsulation protocol.

```
RouterB(config-if)# encap ppp
```

#Set an IP address.

```
RouterB(config-if)# ip address 1.1.1.2 255.255.255.0
```

#Set the receiver of the PPP LCP extended configuration option.

```
RouterB(config-if)#ppp lcp accept-option
```

ACFC Option Processing

In the following example, the interface Serial 1/3 of Router A is configured to actively send the ACFC option during PPP LCP negotiation, enable ACFC, and reply with an ACK response to the PPP LCP configuration request packet that carries the ACFC option and that is sent from the peer end.

Router A:

```
Ruijie# config terminal
```

#Set a host name.

```
Ruijie(config)# hostname RouterA
```

#Enter the interface configuration mode.

```
Ruijie(config)# interface Serial 1/3
```

#Set the encapsulation protocol.

```
RouterA(config-if)# encap ppp
```

#Set an IP address.

```
RouterA(config-if)# ip address 1.1.1.1 255.255.255.0
```

#Configure the function of actively sending the ACFC option.

```
RouterA(config-if)#ppp acfc local request
```

#Configure the function of accepting the PPP LCP configuration request packet that carries the ACFC option.

```
RouterA(config-if)#ppp acfc remote apply
```

PFC Option Processing

In the following example, the interface Serial 1/3 of Router A is configured to actively send the PFC option during PPP LCP negotiation, enable PFC, and reply with an ACK response to the PPP LCP configuration request packet that carries the PFC option and that is sent from the peer end.

Router A:

```
Ruijie# config terminal
```

#Set a host name.

```
Ruijie(config)# hostname RouterA
```

#Enter the interface configuration mode.

```
Ruijie(config)# interface Serial 1/3
```

#Set the encapsulation protocol.

```
RouterA(config-if)# encap ppp
```

#Set an IP address.

```
RouterA(config-if)# ip address 1.1.1.1 255.255.255.0
```

#Configure the function of actively sending the PFC option.

```
RouterA(config-if)#ppp pfc local request
```

#Configure the function of accepting the PPP LCP configuration request packet that carries the PFC option.

```
RouterA(config-if)#ppp pfc remote apply
```

PPP Troubleshooting

First, use the **show interface serial** *slot-number/interface-number* command to check the interface status. A synchronous interface may have four statuses. Take the serial 1/0 as an example;

Status display	Fault description
serial 1/0 is administratively down, line protocol is down	The interface is shut down by someone.
serial 1/0 is down, line protocol is down	The interface is not activated, or the physical interface has not turned Up
Serial 1/0 is up, line protocol is up	The interface is available for data transmission.
serial 1/0 is up, line protocol is down	The interface has been activated but the link negotiation fails.

Interface Cannot be Up

Firstly: Remove the fault of manual shutdown of the interface.
Secondly: Check the physical layer causes. Run the show interface serial1/0 command to check the interface status. All the physical layer parameters (DCD, DTR, DSR, CTS, RTS) shall be up. If some one is down, verify whether the related V.35 or V.24 cable has problem or not.
Finally: If the cable of the interface is DTE (the connector connected with the line device is a fame-end cable) but the DCD is down, verify whether the line handshake of the Modem connected with the cable is successfully. If so, the DCD or LINE indicator on the Modem shall be always on.

Link Protocol Cannot be Up

The interface is up is the prerequisite for Line Protocol Up. Therefore, first remove the interface down fault in troubleshooting the fault of the link protocol cannot be up. Then, verify the data receiving/transmitting of the link layer: Run the **show interface serial *slot-number/interface-number*** command to note the numbers of Packet Input and Packet Output. If there is no Input message, the peer router may be shut down or the transmission of the peer router has problem.
Run the **Clear Count serial *slot-number/interface-number*** command for a period and note the number of Interface reset. If any, it indicates the local router transmission has problem.
To prevent the loopback problem with the line: Run the **show interface serial *slot-number/interface-number*** command to check whether there is the prompt "Loopback is set". Use the **debug PPP packet** command to check whether the PPP negotiation matches the reply of the peer router Magic Number. If yes, the line may be in the loopback status, causing the link cannot be up. It may also possible that the link is up but the peer cannot be pinged through.
If the data transmission/receiving of the line is normal but the protocol settings of the line do not match, such as HDLC protocol set for the peer router, run the **debug PPP packet** command to view the prompt for protocol type. If so, the line protocol cannot be up.
The fault that the link protocol cannot be up is related with the negotiation parameters. If the line negotiation needs the CHAP or PAP authentication, it is necessary to ensure correct username and password, which can be verified by using the **debug ppp packet** or **debug ppp negotiation** command.

Link is Up but Ping Failed

The status of link up is based on the successful LCP negotiation. If the interface has been configured with an IP address, the Line protocol may also be up but there is no prompt "IPCP Open". As a result, if the IP address of the peer WAN interface cannot be pinged through, remove the fault of the IPCP negotiation failure. If the peer WAN interface can be pinged through but the IP addresses inside the peer LAN cannot be pinged through, troubleshoot from the routing table first. For the related routing configuration, see IP Routing Protocol Configuration Guide.

DLDP Configuration

Overview

Based on the SDH platform, the MSTP supports access, processing, and transmission of multiple services such as TDM, ATM, and Ethernet and provides multi-service nodes with a the unified network management system. The Ethernet access mode is commonly used at user access points. However, link keep-alive protocols are not available on Ethernet, causing exceptions where the link protocol status is normal whereas lines are disonnected when MSTP networks are accessed in Ethernet access mode. In this case, route convergence is slow and faults are more difficult to locate.

The major procedure for device link detection falls into the following stages:

3) Initialization

When DLDP is enabled on the interface, the status of DLDP changes to the initialization state, and then an ARP request is sent to obtain the MAC address of the peer device. If DLDP cannot obtain the MAC address of the peer device, DLDP remains in the initialization stage. In this case, if the DLDP function is disabled, the DLDP status changes to deleted. After the MAC address of the peer device is obtained, the DLDP status changes to link succeeded.

4) Link succeeded status

In this state, a DLDP link detection request can be sent to detect line connectivity. If a DLDP response is received, the corresponding interface is marked as UP. If no response is received, requests are sent until the number of requests exceeds the maximum allowed number of requests. In this case, the link is marked as failed and the DLDP status changes to initialization. If this function is deleted during this process, the DLDP status changes to deleted.

5) Deleted status

In the deleted state, the interface status is not analyzed by the link detection function and remains consistent with the physical channel status.

Configuring Device Link Detection

Task List

Configuring the Ethernet Link Detection Function

This command can be configured on the Ethernet port only. By default, this function is disabled. To activate this function, run the following command:

Command	Function
Ruijie(config-if)#**dldp** *ip [nexthopip]*	Activates the link detection protocol.

Note

1. This function is implemented using ICMP ECHO packets. The ICMP response function needs to be enabled on the peer device.
2. The precondition for enabling this function is that the interface is in the UP state.
3. After this function is enabled, the IP address of the interface cannot be modified when the interface is in the down state .
4. In the case of detection across network segments, the next-hop IP address needs to be configured. For example, If the local interface IP address is 10.1.1.1 and 30.1.1.1 needs to be detected through the 20.1.1.2 gateway, the next-hop IP address 20.1.1.2 needs to be configured.

Configuring the Interval

Setting heartbeat intervals can change the frequency of handshake packet sending for link detection.

Command	Function
Ruijie(config-if)# **dldp** ip **interval** *val*	Sets the interval for device link detection.

Configuring Retry Times

Command	Function
Ruijie(config-if)# **dldp** ip **retry** *val*	Sets the threshold of error times during device link detection.

Configuring Resume Times

Command	Function
Ruijie(config-if)# **dldp** ip **resume** *val*	Sets the resumption threshold of the device link. The threshold indicates the times for receiving continuous DLDP detection packet responses before the link status changes from DOWN to UP. The resumption time is related to the interval for sending link detection packets set by running the **dldp ip interval** command. The line resumption time can be calculated using the following formula: Line resumption time = Resume times x Time configured by using the **dldp ip interval** command.

Note

This function is used to avoid oscillation of device links. For example, if link status changes between connnected and disonnected during the link status detection using the **ping** command, continuous oscillation occurs on the link (Link status changes continuously between UP and DOWN or ARP is continuously switched.). This problem can be avoided by setting a greater resume value. Link status changes from DOWN to UP only when the number of detection packet responses received by the link reached the threshold set by using the **resume** command.

Clearing the Records of the Times When DLDP Status Changes Between UP and DOWN

Command	Function
Ruijie(config-if)# **clear-dldp***[all] [ip [nexthopip]]*	Ruijie routers can record the times when DLDP protocol status changes between UP and DOWN. The **clear-dldp** command can be used to clear the recorded times and start the new recording.

Note

1. The **clear-dldp all** command can be used to clear the recorded times when DLDP status changes between UP and DOWN on an interface within a period of time and to start the new recording from 0.
2. The **clear-dldp ip [nexthopip]** command can be used to clear the recorded times when link status changes between UP and DOWN on a specified interface within a period of time and to start the new recording from 0.

Checking the Times When DLDP Status Changes Between UP and DOWN Within a Period of Time

Command	Function
Ruijie(config-if)# **show dldp** interface *[ ] [FastEthernet/GigabitEthernet number]*	Displays the times when DLDP status changes between UP and DOWN within a period of time.

Note

1. The **show dldp interface** command can be used to check the times when DLDP status changes between UP and DOWN on all interfaces and view the time when the recording starts.
2. The **show dldp interface FastEthernet/GigabitEthernet number** command can be used to check the recorded times when protocol status changes between UP and DOWN on an Ethernet interface and view the time when the recording starts.

BFD Configuration

Understanding BFD

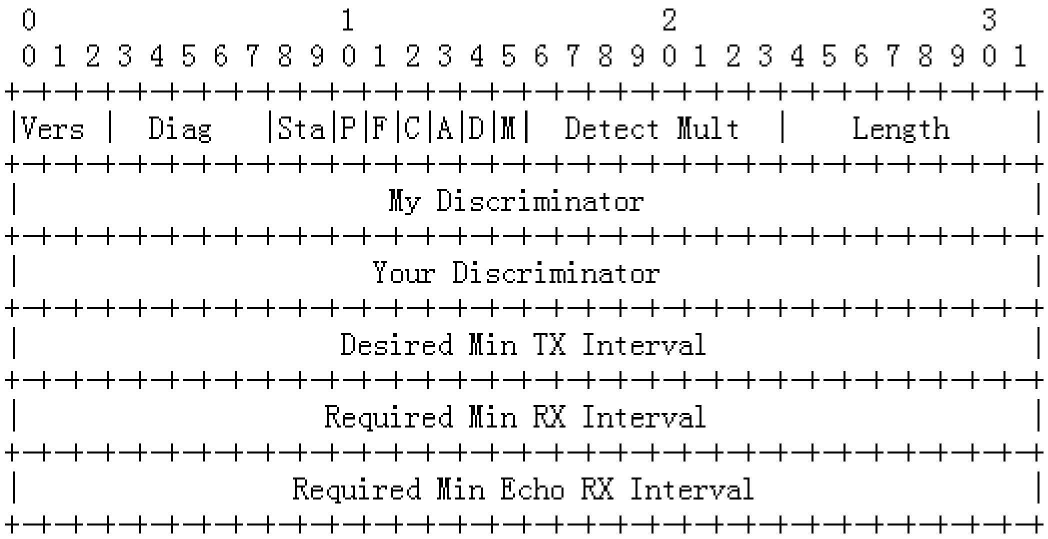
Overview

Bidirectional forwarding detection (BFD) provides low-overhead, short-duration detection of the connectivity in the forwarding path between adjacent routers. The fast detection of failures in the forwarding path speeds up enabling the backup forwarding path and improves the network performance.

BFD Packet Format

The two types of BFD packets are control packets and echo packets. The local end sends echo packets to the peer, which returns the received echo packets back without processing. Therefore, no BFD echo packet format is defined. Only BFD control packet format is defined. There are two versions for the BFD control packet: version 0 and version 1. By default, the BFD session establishment adopts the version 1. However, if one end receives the version 0 control packets from the peer, the default version 1 automatically switches to version 0 to establish the BFD session. You can use the **show bfd neighbors** command to view the version member. The format of the version 1 packet is shown as follows:

Figure 1 Format of BFD control packets (version 1)



```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Vers |  Diag   |Sta|P|F|C|A|D|M|  Detect Mult  |    Length     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       My Discriminator                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Your Discriminator                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Desired Min TX Interval                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Required Min RX Interval                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Required Min Echo RX Interval                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Field	Description
Vers	BFD protocol version. Currently, the value is **1**.
Diag	The following are causes of the latest switchover from the UP state to other states: **0**: indicates no diagnosis. **1**: indicates the control timeout detection. **2**: indicates the echo function failure. **3**: indicates that the neighbor advertising session is down.

Field	Description
	4: indicates that the forwarding plane is reset. **5**: indicates that the channel is invalid. **6**: indicates that the connection channel is invalid. **7**: indicates that administration is down.
Sta	Local status of the BFD. Including: **0**: AdminDown **1**: agent down **2**: agent Init **3**: agent UP
P	When a parameter changes, the sender places this flag in a BFD packet. The receiver must immediately respond the packet.
F	The packet must have the F flag set for responding to the packet with the P flag set.
C	Forward/control separation flag. Once this flag is set, the change of the control plane does not affect the BFD. For example, if the control plane deploys OSPF, the BFD continues with link status detection when OSPF restarts or performs a graceful restart (GR).
A	Authentication flag. If this flag is set, sessions need to be authenticated.
M	Used in point-to-multipoint in the future. Currently, the value must be set to **0**.
Detect Mult	Detection timeout multiples. This flag is used by the detector to compute the timeout duration.
Length	Packet length
My Discriminator	Discriminator used by the BFD session to connect to the local end
Your Discriminator	Discriminator used by the BFD session to connect to the remote end
Desired Min Tx Interval	Minimum BFD packet sending interval supported by the local end
Required Min RX Interval	Minimum BFD packet receiving interval supported by the local end
Required Min Echo RX Interval	Minimum echo packet receiving interval supported by the local end. If the local end does not support the echo function, set the value to 0.
Auth Type	Authentication types (optical), including: Simple Password Keyed MD5 Meticulous Keyed MD5 Keyed SHA1 Meticulous Keyed SHA1
Auth Length	Authentication data length
Authentication Data	Authentication data area

Note

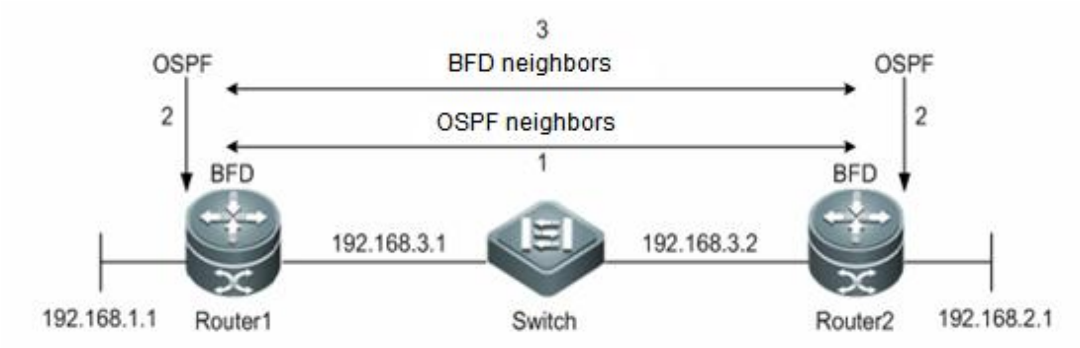
From version 10.3(4b3), the RGOS supports version 1 and version 0 packets. By default, session initiation packets adopt version 1. If one end receives version 0 control packets from the peer, the default version 1 automatically switches to version 0 to establish the session.

BFD Operation Mechanism

The BFD detection mechanism is independent from the applied interface media type, encapsulation format mad, associated upper-layer protocols such as OSPF, BGP, and RIP. The BFD establishes a session between adjacent routers enables the route protocols to re-calculate the route table by rapidly sending the detection fault to the running route protocols and decreases the network convergence time sharply. The BFD cannot discover the neighbors, so it needs the upper-layer protocols to notify the neighbors of which the session is established.

The following figure shows that two routers are connected through a L2 switch. The two routers runs OSPF and BFD.

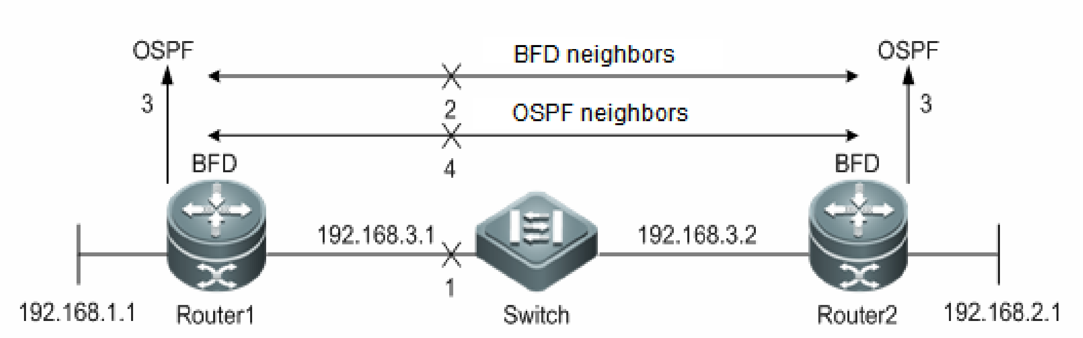
Figure 2 BFD session establishment



The BFD session establishment process is as follows:

6) OSPF discovers neighbors and establishes neighbor relationships.
7) OSPF notifies BFD of establishing the session with the neighbors.
8) BFD establishes the session with the neighbors.

Figure 3 BFD fault detection process



The BFD fault detection process is as follows:

1) Step 1: A link communication failure between Router1 and Router2 occurs.
2) Step 2: BFD session between the Router1 and Router2 detects the fault.
3) Step 3: BFD notifies the fault of the OSPF reachability to the forwarding path of the neighbor.
4) Step 4: OSPF deals with the process of the neighbor Down. If the backup forwarding path exists, and the protocol convergence is performed and the backup forwarding path is enabled.

Related Protocols and Regulations

The related BFD protocols and regulations are:

- draft-ietf-bfd-base-09: Bidirectional Forwarding Detection
- draft-ietf-bfd-generic-05: Generic Application of BFD
- draft-ietf-bfd-mib-06: Bidirectional Forwarding Detection Management Information Base
- draft-ietf-bfd-v4v6-1hop-09: BFD for IPv4 and IPv6 (Single Hop)
- draft-ietf-bfd-multihop-07: BFD for IPv4 and IPv6 (Multihop)
- draft-ietf-bfd-mpls-07: BFD For MPLS LSPs

☑ Currently, no version supports draft-ietf-bfd-mib-06.

BFD Features

This section describes the BFD features.

BFD Session Establishment Mode

The BFD session is established in the following modes:

- Active Mode: Before a session is established, BFD actively sends the BFD control packets regardless of whether any BFD control packet is received from the peer.

BFD Detection Mode

Asynchronous Mode

In the asynchronous mode, the BFD control packets are sent periodically among the systems. If one system receives no BFD control packet from the peer within the BFD interval, the BFD session will be down.

Echo Mode

The local system sends the BFD echo packet periodically. The peer system loops back the echo packet via the forwarding channel. The BFD session will be down if the continuous echo packets are not received within the detection interval. The echo mode can be co-used with the above-mentioned two detection modes. In the echo mode, the packets are forwarded back via the forwarding panel of the peer system rather than the control panel, reducing the delay and speeding up the fault detection in comparison to the control packet sending. In the asynchronous mode, the control packet sending will be decreased with the echo function enabled, for the echo function processes the detection. The echo function must be enabled in the BFD session; otherwise the echo function will be invalid.

☑ Only BFD session version 1 supports the BFD echo function

Caution The **no ip redirects** command must be executed to disable the redirect function of the IP packets and the **no ip deny land** command must be executed to disable the function of anti-attack of the Land-based DDOS before configuring the echo mode.

BFD Session parameters

- BFD session parameters (including Desired Min Tx Interval, Required Min RX Interval, and Detect Mult) must be configured on interfaces at both ends. Otherwise, BFD sessions cannot be created.
- During BFD session creation, interfaces at both ends will negotiate BFD session parameters and accordingly detect the session.
- If BFD session parameters are revised after BFD session creation, interfaces at both ends re-initiate the negotiation. During revision, the BFD session remains in the UP state.

BFD for Dynamic Route Protocols

Configuring BFD for the route protocols improves the convergence performance of the protocol by taking advantages of the faster fault detection of the BFD in comparison to the HELLO mechanism of the protocol. Generally, the fault detection time can be decreased to less than 1s.

Make sure that the BFD for corresponding protocol is enabled on all BFD session neighbors, or the BFD session cannot be established. However, if the dynamic route protocol or other applications have already notify the BFD of establishing the session with the corresponding neighbor, the BFD for this protocol is established.

BFD for Static Route

Configuring BFD for static route prevents the static route from being the forwarding path when the router selects the routing under the circumstances that the configured static route is unreachable. It can rapidly switch to the backup forwarding path if the backup forwarding path exists.

Being different from the dynamic route protocol, the static route protocol has no mechanism of discovering the neighbor. Therefore, when configuring the BFD for static route, the reachability of the next-hop of the static route is dependent on the BFD session state. If the BFD session detects the fault, which means that next-hop of the static route is unreachable; the static route cannot be installed into the RIB. Make sure that the BFD for static route is enabled on all BFD session neighbors, or the BFD session cannot be established. However, if the dynamic route protocol or other applications have already notify the BFD of establishing the session with the corresponding neighbor, the BFD for static route is enabled..

If the BFD session is removed from the peer in the process of the BFD session establishment, the BFD session is down. Under this circumstance, the static route forwarding shall be ensured.

BFD for PBR

Configuring BFD for PBR prevents the PBR from being the forwarding path when the router selects the routing under the circumstances that the configured PBR is unreachable. It can rapidly switch to the backup forwarding path if the backup forwarding path exists.

The method of BFD for PBR is similar to the BFD for static route. If the BFD session detects the fault by following the forwarding path of the specified neighbor, the PBR will be notified of the unreachability to the corresponding next-hop. The PBR reaching the next-hop is ineffective.

Make sure that the BFD for PBR is enabled on all BFD session neighbors, or the BFD session cannot be established. However, if the dynamic route protocol or other applications have already notify the BFD of establishing the session with the corresponding neighbor, the BFD for PBR will be enabled automatically.

If the BFD session is removed from the peer in the process of the BFD session establishment, the BFD session will be down. And under this circumstance, the PBR forwarding shall be ensured.

BFD for VRRP

BFD for VRRP configuration can replace the HELLO mechanism of VRRP itself to realize the fast detection of running state of the master and backup routers and improve the network performance. Generally, the time of failure detection can be shortened to less than 1s.

Make sure that the BFD for VRRP is enabled on the router at both ends, or the BFD session cannot be established. However, if the dynamic route protocol or other applications have already notify the BFD of establishing the session with the corresponding neighbor, the BFD for VRRP will also be configured.

VRRP can also use BFD to follow the specified neighbor. If the BFD session detects the fault of the forwarding path to the neighbor, it will reduce the VRRP priority automatically and trigger the switchover between the master and backup routers. The BFD can be established only when the dynamic route protocol or other applications notify BFD of establishing the session with corresponding neighbor.

BFD Supports to Change the State of Layer3 Interfaces

The BFD supports the function of changing the L3 interface status. In the configuration mode, you can run the **bfd bind peer-ip** command to detect the direct connection address of a specified L3 interface. BFD session status established through this command will generate the corresponding interface BFD status, such as BFD-DOWN/BFD-UP. With this function commonly used in various FRRs, the BFD is used to detect the interface status and perform FRR.

BFD for MPLS-LSP

The BFD for MPLS indicates that the Label Switched Path (LSP) performs fast detection for neighbors through the BFD. The following detection modes are supported:

- Configuring the BFD for static LSP
- Configuring the BFD for LSPs generated by LDP
- Configuring the BFD for LSP reverse links by using IP addresses

BFD for VRF

The BFD supports VPN Routing and Forwarding (VRF) and detects the connectivity of the forwarding path between the Provider Edge (PE) and the Customer Edge (CE).

BFD for Interfaces

Switches: BFD configuration is allowed only on Routed Port and SVI, not on a L3 AP. If the SVI member interface is a L2 AP, BFD session creation fails on the member interface.

Routers: BFD configuration is allowed on synchronous interfaces, asynchronous interfaces, ATM, serial interfaces, frame relay, POS, CPOS, Ethernet interfaces and child interfaces, E1, channelized ATM, and channelized CPOS.

Configuring BFD

Default configurations for the BFD

Function	**Defaults**
BFD session creation mode	Active mode. It cannot be set.
BFD detection mode	Asynchronous mode. The echo function is enabled by default.
BFD session parameter	No default value. It must be set.
BFD authentication method	Disabled. It cannot be set.
BFD for dynamic route protocol	Disabled
BFD for the static route	Disabled
BFD for PBR	Disabled
BFD for VRRP	Disabled
BFD for VRF	Disabled
BFD for MPLS-LSP	Disabled

Configuring BFD Session Parameters

The BFD session parameter has no default value and must be configured. To configure BFD session parameters, run the following command in turn.

Command	**Function**
Ruijie>**enable**	Enters privileged mode.
Ruijie# **configure terminal**	Enters global mode.
Ruijie(config)# **interface** *type number*	Enters interface configuration mode.
Ruijie(config-if)# **bfd interval** *milliseconds* **min_rx** *milliseconds* **multiplier** *interval-multiplier*	Configures BFD session parameters for a specified interface. **interval** *milliseconds*: indicates the minimum sending interval. The unit is millisecond. **min_rx** *milliseconds*: indicates the minimum receiving interval. The unit is millisecond. **multiplier** *interval-multiplier*: indicates the detection timeout multiples.
Ruijie(config-if)# **end**	Exits interface configuration mode and restores privileged mode.

To delete BFD session parameter configurations, run the **no bfd interval** command in interface configuration mode.

The following example shows how to configure BFD session parameters on the Routed Port FastEthernet 0/2.

```
Ruijie# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# interface fastEthernet 0/2
Ruijie(config-if)# bfd interval 100 min_rx 100 multiplier 3
```

Caution

The difference of the bandwidth transmitted on different interfaces must be considered when configuring the parameters. If the minimum sending and receiving intervals are too low, it may result in the oversized bandwidth of the BFD and affect data transmission.

Switches cannot be configured on the L3 AP interface.

Configuring the BFD Echo Function

The session status is not affected if the echo function is enabled after the BFD session is created. After the echo function is disabled, no echo packet is sent and the forwarding plane ceases to receive echo packets. To configure the BFD echo function, run the following commands in turn.

Command	Function
Ruijie>**enable**	Enters privileged mode.
Ruijie# **configure terminal**	Enters global mode.
Ruijie(config)# **interface** *type number*	Enters interface configuration mode.
Ruijie(config-if)# **bfd echo**	Enables the BFD echo function.
Ruijie(config-if)# **end**	Exits interface mode and returns to privileged mode.

To disable the BFD echo function, run the **no bfd echo** command in interface mode.

The following example shows how to configure the BFD echo function on the Routed Port FastEthernet 0/2:

```
Ruijie# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# interface fastEthernet 0/2
Ruijie(config-if)# bfd echo
```

To enable the BFD control packet to be sent in a slower frequency after the echo function is enabled in the BFD asynchronous mode, run the following commands in turn.

Command	Function
Ruijie>**enable**	Enters privileged mode.
Ruijie# **configure terminal**	Enters global mode.
Ruijie(config)# **bfd slow-timer** milliseconds	Configures the slow-timer. The default value is 1 second. The value range is from 1000 to 30000 and the

	unit is millisecond.
Ruijie(config)# **end**	Enters global configuration mode.

To restore the default value of the slow-timer, run the **no bfd slow-time** mode in global mode.

The following example shows how to configure the time of the slow-timer to 1400 milliseconds:

```
Ruijie# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# bfd slow-timer 1400
```

Caution

The local end sends the BFD echo packet to the peer, which returns the received packets with processing on the forwarding panel. In this process, the BFD session detection may fail for the peer has been congested resulting in the loss of the echo packets. Under these circumstances, the corresponding QoS policy is necessary to be configured to make sure that the echo packets take the precedence to be processed or the echo function is disabled.

Configuring the BFD UP-Dampening Time

The BFD up-dampening time configuration solves the problem that the BFD session status frequent switches between DOWN and UP due to the line instability, which results in the frequent forwarding path switchover of the associated application (for example, the static route) and the abnormal operation. This feature allows you to configure the required up-dampening time before advertising the session UP state to a related application. To configure the BFD UP-Dampening function, run the following commands in turn.

Command	**Function**
Ruijie>**enable**	Enters privileged mode.
Ruijie# **configure terminal**	Enters global mode.
Ruijie(config)# **interface** *type number*	Enters interface configuration mode.
Ruijie(config)# **bfd up-dampening** *milliseconds*	Configures the up-dampening time.
Ruijie(config)#**end**	Exits global mode

To restore the default value, run the **no bfd up-dampening** command in interface configuration mode.

The following example shows how to configure the BFD up-dampening time as 60,000ms:

```
Ruijie# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# interface fastEthernet 0/2
Ruijie(config-if)# bfd up-dampening 60000
```

Caution

In the configuration process, the parameter configurations for two ends of the BFD session must be consistent. This ensures that applications and protocols associated with the BFD take effect simultaneously, and avoid that unidirectional communication occurs on the forwarding path due to different up-dampening time on the two ends.

The dampening function does not take effect in the BFD for OSPFv3.

Configuring the BFD CPP

The BFD protocol is a sensitive protocol. When the device with BFD function enabled suffers from attack (for example, a large amount of Ping packets attack the device), which lead to the BFD session turbulence, the device can be protected by enabling the BFD protection policy. However, if the BFD function and the BFD protection policy are enabled at the same time, the loss of BFD packets on the attacked device occurs when the packets sent from the last-hop device go through this device, influencing the BFD session establishment between the last-hop device and other devices. This function takes effect only for the switches.

To configure the BFD CPP, run the following commands in turn.

Command	Function
Ruijie>**enable**	Enters privileged mode.
Ruijie# **configure terminal**	Enters global mode.
Ruijie(config)# **bfd cpp**	Enables the BFD CPP.
Ruijie(config)# **end**	Exits global configuration mode.

By default, the BFD CPP is enabled. To disable the BFD CPP, run the **no bfd cpp** command in global mode.

The following example shows how to enable the BFD CPP.

```
Ruijie# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# bfd cpp
```

Configuring the BFD for RIP

RIP sends the route updating information periodically. A route is invalid and RIP cannot rapidly respond to the link failure when no route updating information is received within the specified time.

After enabling the BFD for RIP, the BFD session will be established for the RIP route information source (the source address for RIP route updating packet). Once BFD detects that a neighbor is invalid, RIP route information will directly be in the invalid state and not join in the route forwarding no longer. The convergence time can be decreased from 180s (the default RIP timer) to less than 1s.

Run the **bfd all-interfaces** command to enable BFD for RIP applications on all interfaces. Or run the **ip rip bfd** [**disable**] command in interface configuration mode to enable or disable to allow BFD for RIP applications on specified interfaces.

Command	Function
Ruijie>**enable**	Enters privileged mode.
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **router rip**	Enters router configuration mode.
Ruijie(config-router)# **bfd all-interfaces**	Allows the BFD for RIP on all interfaces.
Ruijie(config-router)# **exit**	Exits router configuration mode and returns to global configuration mode. (Optional)
Ruijie(config)# **interface** *type number*	Enters interface configuration mode. (Optional)

Command	**Function**
Ruijie(config-if)#**ip rip bfd** [**disable**]	Allows or prohibits the BFD for RIP on specified interfaces. (Optional)
Ruijie(config-if)#**end**	Exits interface configuration mode. (Optional)
Ruijie#**show bfd neighbors** [**details**]	Displays the BFD session establishment information and whether RIP is for the specified session. (Optional)

To disable to allow the BFD for RIP applications, run the **no bfd all-interfaces** in Router mode.

The following example shows how to enable the BFD for RIP on all interfaces excluding the FastEthernet 0/2:

```
Ruijie# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# router rip
Ruijie(config-router)# bfd all-interfaces
Ruijie(config-router)# exit
Ruijie(config)# interface FastEthernet 0/2
Ruijie(config-if)# ip rip bfd disable
Ruijie(config-if)#end
```

Caution

The route information sources (source address for RIP route updating packet) of two devices with RIP enabled must be in the same network segment to establish the BFD session between adjacent routers.
BFD session parameters must have been configured. Otherwise, BFD session creation fails.
For the non-unnumbered interface, if the neighbor end and the local end are not connected directly, the BFD for IPv4 PBR fails to be enabled.
BFD session creation fails if an interface specified during BFD session creation is different from the actual BFD packet egress or ingress interface due to IP routing.
BFD session creation fails if an interface specified during BFD session creation is different from the actual BFD packet egress or ingress interface.

Configuring the BFD for OSPF

The OSPF protocol dynamically discovers the neighbors by the Hello packets. With BFD for OSPF configured, the BFD session for the neighbors in FULL relationship will be established and the neighbor state will be detected by the BFD mechanism. Once BFD neighbor is invalid, OSPF processes the network convergence. The convergence time could be from 120s(by default, the sending interval of the OSPF Hello packet in non-broadcast network is 30s, which is a quarter of the invalid time for the adjacency router, namely, 120s) to less than 1s.

Run the **bfd all-interfaces** command to enable BFD for RIP applications on all interfaces. Or run the **ip ospf bfd** [**disable**] command in interface configuration mode to enable or disable to allow the BFD for RIP applications on specified interfaces.

Command	**Function**
Ruijie>**enable**	Enters privileged mode.

Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **router ospf** *process-id*	Enters router configuration mode.
Ruijie(config-router)# **bfd all-interfaces**	Allows the BFD for RIP on all interfaces. Prohibits the BFD for RIP on all interfaces by running the **no** command.
Ruijie(config-router)# **exit**	Exits router configuration mode and returns to global configuration mode. (Optional)
Ruijie(config)# **interface** *type number*	Enters interface configuration mode. (Optional)
Ruijie(config-if)#**ip ospf bfd** [**disable**]	Allows or prohibits the BFD for RIP on specified interfaces. (Optional)
Ruijie(config-if)#**end**	Exits interface configuration mode. (Optional)
Ruijie#**show bfd neighbors** [**details**]	Displays the BFD session establishment information and whether OSPF is for the specified session. (Optional)
Ruijie#**show ip ospf**	Displays whether OSPF is for the specified session. (Optional)

To disable to allow the BFD for RIP applications, run the **no bfd all-interfaces** in Router mode.

The following example shows how to enable the BFD for OSPF on all interfaces excluding the FastEthernet 0/2:

```
Ruijie# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# router ospf 123
Ruijie(config-router)# bfd all-interfaces
Ruijie(config-router)# exit
Ruijie(config)# interface FastEthernet 0/2
Ruijie(config-if)# ip rip bfd disable
Ruijie(config-if)#end
```

☑ 10.3(4b3) or 10.3(5) do not support the BFD for OSPFv3.

⚠ Caution

BFD session parameters must have been configured. Otherwise, BFD session creation fails.

BFD session creation fails if an interface specified during BFD session creation is different from the actual BFD packet egress or ingress interface due to IP routing.

BFD session creation fails if an interface specified during BFD session creation is different from the actual BFD packet egress or ingress interface.

The OSPFv2/OSPFv3 virtual link does not support the BFD monitoring.

Configuring Association Between IS-IS and BFD

The Intermediate System to Intermediate System (IS-IS) protocol dynamically discovers a neighbor via hello packets. When the BFD function is enabled, the device running the IS-IS protocol establishes a BFD session for a neighbor in the UP state, and detects the status of the neighbor via the BFD mechanism. Once the BFD neighbor fails, the device running the IS-IS protocol immediately performs network convergence. The network convergence period can be reduced from 30s to 1s. By default, the transmission interval of point-to-point hello packets is 10s, and the failure period of the neighbor device is 30s (3 times the transmission interval).

Run the **bfd all-interfaces** command to enable association between IS-IS and BFD on all interfaces. Run the **isis bfd [disable]** command in interface configuration mode to enable or disable association between IS-IS and BFD on a specified interface.

Command	Function
Ruijie>**enable**	Enters the privileged EXEC mode.
Ruijie# **configure terminal**	Enters the global configuration mode.
Ruijie(config)# **router isis**	Enters the router configuration mode.
Ruijie(config-router)# **bfd all-interfaces**	Enables association between IS-IS and BFD on all interfaces. Runs the **no** form of this command to disable association between IS-IS and BFD on all interfaces.
Ruijie(config-router)# **exit**	(Optical) Exits the router configuration mode and returns to the global configuration mode.
Ruijie(config)# **interface** *type number*	(Optical) Enters the interface configuration mode.
Ruijie(config-if)#**isis bfd** [**disable**]	(Optical) Enables or disables association between IS-IS and BFD on a specified interface.
Ruijie(config-if)#**end**	(Optical) Exits the interface configuration mode.
Ruijie#**show bfd neighbors** [**details**]	(Optical) Displays BFD session information and checks whether IS-IS is associated with a specified session.
Ruijie# **show isis neighbors detail**	(Optical) Displays information about whether IS-IS is associated with a specified session.

Run the **no bfd all-interfaces** command in router configuration mode to disable association between IS-IS and BFD on all interfaces.

Configuration example:

#Enable association between IS-IS and BFD on interfaces except FastEthernet 0/2.

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# router isis
Ruijie(config-router)# bfd all-interfaces
Ruijie(config-router)# exit
Ruijie(config)# interface FastEthernet 0/2
Ruijie(config-if)# isis bfd disable
Ruijie(config-if)#end
```

Caution

BFD session parameters must be configured before enabling association between IS-IS and BFD.

IP routing may cause inconsistency between the interface specified by the BDF session neighbor and the actual BDF packet outbound interface. In this case, BDF sessions cannot be established.

If the interface specified for BFD session establishment is inconsistent with the BFD packet inbound interface, BDF sessions cannot be established.

Configuring the BFD for BGP

Being similar to OSPF, by configuring the BFD for BGP, the BGP protocol rapidly detects the faults, realizes the rapid detection of the neighbor relationship and fastens the protocol convergence. By default, the BGP keepalive interval is 60s and the holdtime is 180s. The minimum value of the keepalive interval and holdtime are 1s and 3s respectively. It is slow to detect the neighbor relationship.

A large amount of the packets will be lost on the interface that receives and sends the packets at the fast speed. With the BFD enabled, the holdtime can decrease to less than 1 second.

Run the **neighbor** *ip-address* **fall-over bfd** command to enable the BFD for BGP.

Command	Function
Ruijie>**enable**	Enters privileged mode.
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **router bgp** *as-tag*	Enters router configuration mode.
Ruijie(config-router)#**neighbor** *ip-address* **fall-over bfd**	Selects BFD keywords to indicate the BFD for BGP to detect the faults of specified neighbors.
Ruijie(config-router)#**end**	Exits router configuration mode. (Optional)
Ruijie#**show bfd neighbors** [**details**]	Displays the BFD session establishment information and whether the BGP is associated to the specified session. (Optional)
Ruijie#**show ip bgp neighbors**	Displays whether the BGP is associated to the specified session. (Optional)

To disable the BFD for BGP applications, run the **no neighbor** *ip-address* **fall-over bfd** in Router mode.

The following example shows how to enable the BFD for BGP, and detect the forwarding path with the neighbor 172.16.0.2:

```
Ruijie# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# interface FastEthernet 0/1
Ruijie(config-if)# no switchport
Ruijie(config-if)# ip address 172.16.0.1 255.255.255.0
Ruijie(config-if)# bfd interval 50 min_rx 50 multiplier 3
Ruijie(config-if)# exit
Ruijie(config)# router bgp 44000
Ruijie(config-router)# bgp log-neighbors-changes
Ruijie(config-router)# neighbor 172.16.0.2 remote-as 45000
Ruijie(config-router)# neighbor 172.16.0.2 fall-over bfd
Ruijie(config-router)# end
```

☑ 10.3(4b3) and 10.3(5) support the BFD for BGP only when BGP works in IPv4 address families.

Caution

If BGP establishes the session using the loopback address and enables BFD to detect the neighbors, the outbound interface for the BFD packets will be specified according to the result of IP routing. In this situation, before configuring the BFD for BGP, the **bfd interval** command is necessary to be used to configure the BFD session parameter on the possible outbound interface. Otherwise, it may fail to establish the session. The BFD session cannot be established if the specified interface and the actual incoming interface for the BFD packets are inconsistent.

Configuring the BFD for Static Route

To configure the BFD for Static Route, run the following commands in turn.

Command	Function
Ruijie>**enable**	Enters privileged mode.
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **ip route static bfd** [**vrf** *vrf-name*] *interface-type interface-number gateway* [**souce** *ip-address*]	Configures the BFD for the static route. The parameters *interface-type*, *interface-number* and *gateway* indicate the interface and IP address of a neighbor. For multi-hops, **souce** *ip-address* needs to be configured as the session source IP address. BFD session parameters must be configured before the configuration.
Ruijie(config)#{ **ip route** *prefix mask* {*ip-address* | *interface-type interface-number [ip-address*]}	Ensures the parameters *interface-type*, *interface-number* and *gateway* are consistent with that configured in Step 3 before configuring the BFD for the static route.
Ruijie(config)# **end**	Exits global configuration mode.
Ruijie#**show bfd neighbors** [**details**]	Displays the BFD session establishment information and whether the static route is associated to the specified session. (Optional).

To disable the BFD for the static route, run the **no ip route static bfd** [**vrf** *vrf-name*] *interface-type interface-number gateway* command in interface mode.

The following example shows how to enable the BFD for static route, and detect the forwarding path with the neighbor 172.16.0.2:

```
Ruijie# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
```

```
Ruijie(config)# interface FastEthernet 0/1
Ruijie(config-if)# no switchport (This configuration is unnecessary for routers)
Ruijie(config-if)# ip address 172.16.0.1 255.255.255.0
Ruijie(config-if)# bfd interval 50 min_rx 50 multiplier 3
Ruijie(config-if)# ip route static bfd FastEthernet 0/1 172.16.0.2
Ruijie(config-if)# ip route 10.0.0.0 255.0.0.0 FastEthernet 0/1 172.16.0.2
Ruijie(config-if)# end
```

Configuring the BFD for PBR

To configure the BFD for PBR, run the following commands in turn.

Command	Function
Ruijie>**enable**	Enters privileged mode.
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **route-map** *route-map-name* [**permit** | **deny**] *sequence*	Configures the defined route map to enter route map configuration mode.
Ruijie(config-route-map)#**match ip address** *access-list-number*	Configures the matching access list.
Ruijie(config-route-map)#**set ip next-hop verify-availability** *next-hop-address* { **track** *number* **| bfd** [**vrf** *vrf-name*] *interface-type interface-number gateway* }	Configure the BFD for PBR. The parameters *interface-type*, *interface-number* and *gateway* indicate the interface and IP address of a neighbor. BFD session parameters must be configured before the configuration. The next hop specified by the parameter *next-hop-address* is unreachable if the BFD session detects faults. Run the **no** command to cancel the configuration.
Ruijie(config-route-map)#**exit**	Exits route map configuration mode.
Ruijie(config)# **interface** *type number*	Enters interface configuration mode.
Ruijie(config-if)#**ip policy route-map** *route-map*	Enables the PBR on interfaces.
Ruijie(config-if)#**end**	Exits interface configuration mode.
Ruijie#**show bfd neighbors** [**details**]	Displays the BFD session establishment information and whether the PBR is associated with the specified session. (Optional)
Ruijie#**show route-map**	Displays whether the PBR is associated with the specified session. (Optional)

To disable the BFD for PBR applications, run the **no set ip next-hop verify-availability** [*next-hop-address* [**track** *number*|**bfd** [**vrf** *vrf-name*] *interface-type interface-number gateway*]] command in router-map mode.

The following example shows how to enable the BFD for PBR, and detect the forwarding path with the neighbor 172.16.0.2:

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# route-map Example1 permit 10
Ruijie(config-route-map)# match ip address 1
Ruijie(config-route-map)#set ip next-hop verify-availability 172.16.0.2 bfd FastEthernet 0/1
172.16.0.2
Ruijie(config-route-map)#end
Ruijie(config)#interface FastEthernet 0/1
Ruijie(config-if)#no switchport (This configuration is unnecessary for routers)
Ruijie(config-if)#ip address 172.16.0.1 255.255.255.0
Ruijie(config-if)#bfd interval 50 min_rx 50 multiplier 3
Ruijie(config-if)#ip policy route-map Example1
Ruijie(config-if)#exit
```

☑ The BFD for PBRv6 is not supported in v10.3(4b3) and v10.3(5).

Configuring the BFD for VRRP

To enable the BFD for VRRP groups to detect the master and backup routers, run the following commands in turn.

Command	Function
Ruijie>**enable**	Enters privileged mode.
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **interface** *type number*	Enters interface mode.
Ruijie(config-if)#**vrrp** *group-number* **ip** [*ip-address*[**secondary**]]	Creates VRRP groups and virtual IP addresses. *ip-address* indicates the IP address of the specified neighbor.
Ruijie(config-if)# **vrrp** *group-number* **bfd** *ip-address*	Configures the BFD for all VRRP groups. *ip-addess* indicates the neighbor IP address.
Ruijie(config-if)# **end**	Exits interface configuration mode.
Ruijie#**show bfd neighbors** [**details**]	Displays the BFD session establishment information and whether the VRRP is associated with the specified session. (Optional).
Ruijie#**show vrrp**	Displays whether the VRRP is associated with the specified session. (Optional)

To disable the BFD for VRRP groups to detect the master and backup routers, run the **no vrrp** *group-number* **bfd** command in interface mode.

The following example shows how to enable the BFD for VRRP, and detect the forwarding path between the master and slave routers:

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)#interface FastEthernet 0/1
Ruijie(config-if)#no switchport (This configuration is unnecessary for routers)
Ruijie(config-if)#ip address 192.168.201.11 255.255.255.0
Ruijie(config-if)#bfd interval 50 min_rx 50 multiplier 3
Ruijie(config-if)#vrrp 1 priority 120
Ruijie(config-if)#vrrp 1 ip 192.168.201.1
Ruijie(config-if)#vrrp 1 bfd 192.168.201.12
Ruijie(config-if)#end
```

To enable the specified VRRP group to track the IP address of the specified neighbor through the BFD, run the following commands in turn.

Command	**Function**
Ruijie>**enable**	Enters privileged mode.
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **interface** *type number*	Enters interface configuration mode.
Ruijie(config-if)#**vrrp** *group-number* **ip** [*ip-address*[**secondary**]]	Creates VRRP groups and virtual IP addresses on specified interfaces.
Ruijie(config-if)# **vrrp** *group-number* **track bfd** *interface-type interface-number ip-addess* [*priority*]	Configures the specified VRRP group to track the IP address of the specified neighbor through the BFD. Run the **no** command to cancel the configuration.
Ruijie(config-if)# **end**	Exits interface configuration mode.
Ruijie#**show bfd neighbors** [**details**]	Displays the BFD session establishment information and whether the VRRP is associated with the specified session. (Optional).
Ruijie#**show vrrp**	Displays whether the BFD is for VRRP is enabled to track the IP address of the specified neighbor. (Optional).

To disable the specified VRRP group to track the IP address of the specified neighbor through the BFD, run the no **vrrp** *group-number* **track bfd** *interface-type interface-number ip-addess* command in interface mode.

The following example shows how to specify the VRRP to track the specified neighbor 192.168.1.3:.

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)#interface FastEthernet 0/1
Ruijie(config-if)#no switchport (This configuration is unnecessary for routers)
Ruijie(config-if)#ip address 192.168.1.1 255.255.255.0
Ruijie(config-if)#bfd interval 50 min_rx 50 multiplier 3
Ruijie(config)#interface FastEthernet 0/2
Ruijie(config-if)#no switchport (This configuration is unnecessary for routers)
Ruijie(config-if)#ip address 192.168.201.17 255.255.255.0
Ruijie(config-if)#vrrp 1 priority 120
Ruijie(config-if)#vrrp 1 ip 192.168.201.1
Ruijie(config-if)#vrrp 1 track bfd FastEthernet 0/1 192.168.1.3 30
Ruijie(config-if)#end
```

Configuring BFD for Changing the State of Layer 3 Interfaces

Generally, it takes a long time for link communication failure or link failure to change the interface state. For various FRRs relying on interface state, high-performance switchover cannot be achieved. Therefore, BFD is generally associated with the layer-3 interface state to realize fast detection of interface state. Execute the following configurations to associate BFD and layer 3 interface states.

Command	**Function**
Ruijie>**enable**	Enters privileged mode.
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **interface** *type number*	Enters an L3 interface.
Ruijie(config-if)# **bfd bind peer-ip** *ip-address* [**source-ip** *ip-adress*] **process-pst**	Configure the BFD for L3 interfaces. Source-ip specifies the source IP address of a BFD packet for avoiding that the packet is discarded due to uRPF check failure when the uRPF is also used. Process-pst indicates the BFD status of the BFD session generation interface.
Ruijie(config-if)#**end**	Exits interface configuration mode. (Optional).
Ruijie#**show bfd neighbors** [**details**]	Displays the BFD session establishment information and whether the interface is associated with the specified session. (Optional).

To disable the BFD for interfaces, run the **no bfd bind peer-ip** *ip-address* in configuration mode.

The following example shows how to enable the BFD for FastEthernet 0/2.

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# interface FastEthernet 0/2
Ruijie(config-if)#no sw (This configuration is unnecessary for routers)
Ruijie(config-if)#ip address 1.1.1.1 255.255.255.0
Ruijie(config-if)#bfd bind peer-ip 1.1.1.2 source-ip 1.1.1.1 process-pst
Ruijie(config-if)#end
```

Configuring the BFD for MPLS

The BFD for MPLS indicates that the BFD performs rapid detection on the LSP on the MPLS network to improve MPLS network reliability.

Configuring the BFD Detection for the Static LSP

Command	**Function**
Ruijie>**enable**	Enters privileged mode.
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **bfd bind static-lsp peer-ip** *ip-address* **source-ip** *ip-address* [**local-discriminator** *discr-value* **remote-discriminator** *discr-value*] [**process-state**]	Configures the BFD for static LSP. You can configure the homing address, next-hop address, and egress interface of the LSP. If no local discriminator is configured, the system automatically elects the local discriminator. If no remote discriminator is configured, the system learns of the remote discriminator in automatic configuration mode.

Configuring the BFD detection for the Dynamic LSP

Command	**Function**
Ruijie>**enable**	Enters privileged mode.
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie#**mpls router ldp**	Enters LDP configuration mode.
Ruijie(config-mpls-router)# **bfd bind ldp-lsp peer-ip** *ip-address* **nexthop** *ip-address* [**interface** *interface-type interface-number*] **source-ip** *ip-address* [**local-discriminator** *discr-value* **remote-discriminator** *discr-value*] [**process-state**]	Configures the BFD for dynamic LSP. You can configure the homing address, next-hop address, and egress interface of the LSP. If no local discriminator is configured, the system automatically elects the local discriminator. If no remote discriminator is configured, the system learns of the remote discriminator in automatic configuration mode.

Configuring the IP Address Mode for Reverse Link Detection in BFD for the LSP

Command	**Function**
Ruijie>**enable**	Enters privileged mode.
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **bfd bind backward-lsp-with-ip peer-ip** *ip-address* [**vrf** *vrf-name*] **interface** *interface-type interface-number* [**source-ip** *ip-address*] {**local-discriminator** *discr-value* **remote-discriminator** *discr-value*}	Configures the IP address mode for reverse link detection in BFD for the LSP. You can configure the source addresses, homing address, and egress interface of the LSP. Both the local and remote discriminators must be manually configured.

For details about enabling the BFD for MPLS-LSP, see the documents *MPLS-CREF* and *MPLS-SCG*.

Displaying BFD Configuration and Status

BFD provides following display commands for checking various configuration and running information. The following table describes functions of commands.

Command	**Function**
show bfd neighbors [**vrf** *vrf-name*] [**ipv4** *ip-addess* [**details**]| **ipv6** *ipv6-addess* [**details**] |**client** {**bgp**|**ospf**|**rip**|**vrrp**|**vrrp-balance** |**ldp-lsp**|**static-lsp**|**backward-lsp-with-ip**|**static-route**| **pbr**|**pst**} **[ipv4** *ip-addess* [**details**] | **ipv6** *ipv6-addess* [**details**]| **details**]]	Displays the BFD session information. For details, see the description about session display fields in Figure 4.
show vrrp	Displays information about enabling the BFD for VRRP.
show route-map	Displays information about enabling the BFD for PBR.
show ip route static bfd	Displays information about enabling the BFD for static route.
show ip bgp neighbors	Displays information about enabling the BFD for BGP.
show ip ospf neighbor	Displays information about enabling the BFD for OSPF.
show ip rip peer	Displays information about enabling the BFD for RIP.

Note The preceding display commands can be configured in any mode except for user mode.

☑ 10.3(4b3), 10.4(1), and later versions support the preceding configurations.

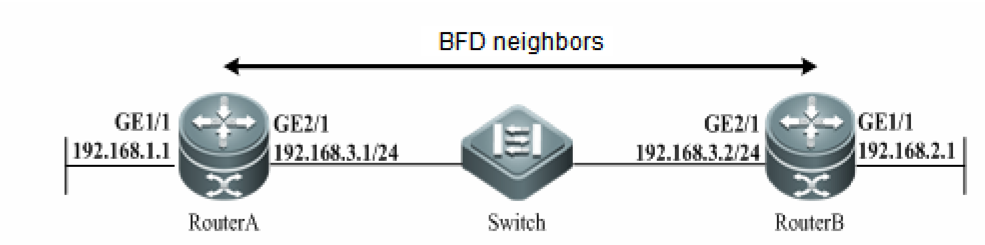
Example of Configuring BFD for RIP

Networking Requirement

Router A and Router B are interconnected through a L2 switch. Both routers run the RIP protocol and enable the BFD for RIP on the interface. After a link failure between Router B and L2 switch occurs, BFD detects the failure and notifies the RIP of the failure, triggering the rapid convergence.

Networking Topology

Figure 4 Topology of configuring BFD for RIP



Configuration Tips

- Router A Configuration

# Configure the Routed Port GE 2/1, the IP address, and the BFD session parameter for Router A.

```
Ruijie# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# interface GigabitEthernet2/1
Ruijie(config-if)# no switchport (This configuration is unnecessary for routers)
Ruijie(config-if)# ip address 192.168.3.1 255.255.255.0
Ruijie(config-if)# bfd interval 200 min_rx 200 multiplier 5
```

Configure the Routed Port GE1/1.

```
Ruijie(config-if)# exit
Ruijie(config)# interface GigabitEthernet1/1
Ruijie(config-if)# no switchport (This configuration is unnecessary for routers)
Ruijie(config)# ip address 192.168.1.1 255.255.255.0
```

# Enable RIP and configure the BFD for RIP to detect the neighbor 192.168.3.2.

```
Ruijie(config-if)# exit
Ruijie(config)# router rip
Ruijie(config-router)# version 2
Ruijie(config-router)# network 192.168.3.0
Ruijie(config-router)# network 192.168.1.0
Ruijie(config-router)# passive-interface GigabitEthernet 2/1
Ruijie(config-router)# bfd all-interfaces
```

- Router B Configuration

# Configure the Routed Port, the IP address, and the BFD session parameter for Router B.

```
Ruijie# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# interface GigabitEthernet 2/1
Ruijie(config-if)# no switchport (This configuration is unnecessary for routers)
Ruijie(config-if)# ip address 192.168.3.2 255.255.255.0
Ruijie(config-if)# bfd interval 50 min_rx 50 multiplier 3
```

# Configure the Routed Port GE 1/1.

```
Ruijie(config-if)# exit
Ruijie(config)# interface GigabitEthernet1/1
Ruijie(config-if)# no switchport (This configuration is unnecessary for routers)
Ruijie(config-if)# ip address 192.168.2.1 255.255.255.0
```

# Enable RIP and configure the BFD for RIP to detect the neighbor 192.168.3.1.

```
Ruijie(config-if)# exit
Ruijie(config-router)# router rip
Ruijie(config-router)# version 2
Ruijie(config-router)# network 192.168.3.0
Ruijie(config-router)# network 192.168.2.0
Ruijie(config-router)# passive-interface GigabitEthernet 2/1
Ruijie(config-router)# bfd all-interfaces
Ruijie(config-router)# end
Ruijie#
```

Verification

- View the BFD session of Router A

```
Ruijie# show bfd neighbors details
OurAddr         NeighAddr      LD/RD   RH  Holdown(mult)  State   Int
192.168.3.1     192.168.3.2    1/2     1       532 (3 )       Up      Ge2/1
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 200000, MinRxInt: 200000, Multiplier: 5
Received MinRxInt: 50000, Received Multiplier: 3
Holdown (hits): 600(22), Hello (hits): 200(84453)
```

```
Rx Count: 49824, Rx Interval (ms) min/max/avg: 208/440/332
Tx Count: 84488, Tx Interval (ms) min/max/avg: 152/248/196
Registered protocols: RIP
Uptime: 02:18:49
Last packet:    Version: 1             - Diagnostic: 0
I Hear You bit: 1       - Demand bit: 0
Poll bit: 0             - Final bit: 0
Multiplier: 3               - Length: 24
My Discr.: 2            - Your Discr.: 1
Min tx interval: 50000 - Min rx interval: 50000
Min Echo interval: 0
```

Field	Description
OurAddr	IP address for the session on the local end
NeighAddr	IP address for the adjacent session
LD/RD	Session ID on the local and remote end
RH/RS	Current status of the session peer end
Holdown(mult)	Time of not receiving the Hello packets on the local end of the session
State	Current session status
Int	Interface number for the session
Session state is UP and using echo function with 50 ms interval	Whether the session is in echo mode and the interval of sending frames. This information is shown only in echo mode.
Local Diag	Diagnosis information of the session
Poll bit	Whether the session configuration is modified.
MinTxInt	Minimum sending interval of the session on the local end
MinRxInt	Minimum receiving interval of the session on the local end
Multiplier	Timeout times detected on the local end
Received MinRxInt	Minimum sending interval of the session on the remote end
Received Multiplier	Timeout times detected on the remote end
Holdown (hits)	Session detection time and the detected timeout times
Hello (hits)	Minimum interval of receiving the Hello packet after the session negotiation
Rx Count	Number of BFD packets received on the local end
Rx Interval (ms) min/max/avg	Minimum/maximum/average interval of receiving the session on the local end
Tx Count	Number of BFD packets sent from the local end

Field	Description
Tx Interval (ms) min/max/avg	Minimum/maximum/average interval of sending the session from the local end
Registered protocols	Type of protocol registered to the session
Uptime	Time of keeping the session UP
Last packet	Last BFD packet received by the local end

- View the BFD session of Router B

```
Ruijie# show bfd neighbors details
OurAddr         NeighAddr      LD/RD   RH  Holdown (mult)  State   Int
192.168.3.2     192.168.3.1    2/1     1       532 (5 )       Up      Ge2/1
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 50000, MinRxInt: 50000, Multiplier: 3
Received MinRxInt: 200000, Received Multiplier: 5
Holdown (hits): 600(22), Hello (hits): 200(84453)
Rx Count: 49824, Rx Interval (ms) min/max/avg: 209/440/332
Tx Count: 84488, Tx Interval (ms) min/max/avg: 153/249/197
Registered protocols: RIP
Uptime: 02:18:49
Last packet:    Version: 1             - Diagnostic: 0
I Hear You bit: 1       - Demand bit: 0
Poll bit: 0             - Final bit: 0
Multiplier: 5               - Length: 24
My Discr.: 1            - Your Discr.: 2
Min tx interval: 200000     - Min rx interval: 200000
Min Echo interval: 0
```

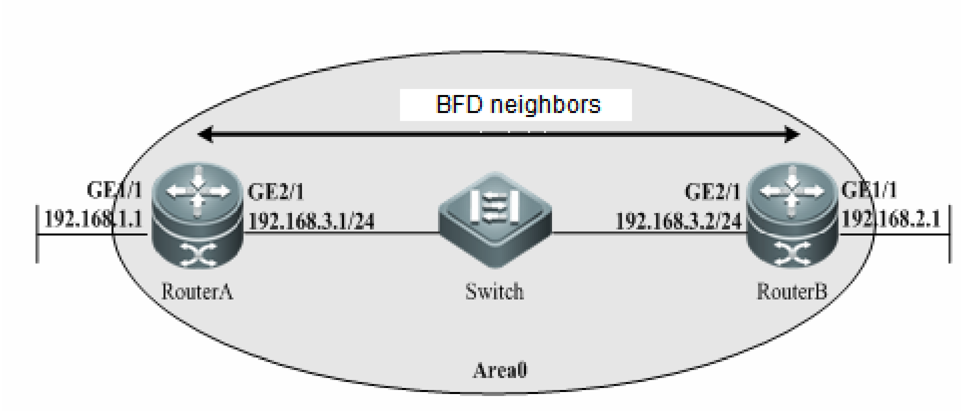
Example of Configuring BFD for OSPF

Networking Requirement

Router A and Router B are interconnected through a L2 switch. Both routers run the OSPF protocol and enable the BFD for OSPF on the interface. After a link failure between Router B and L2 switch occurs, BFD detects the failure and notifies the OSPF of the failure, triggering the rapid convergence.

Networking Topology

Figure 5 Topology of Configuring BFD for OSPF



Configuration Steps

- Router A Configuration

# Configure the Routed Port, the IP address, and the BFD session parameter for Router A.

```
Ruijie# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# interface GigabitEthernet2/1
Ruijie(config-if)# no switchport (This configuration is unnecessary for routers)
Ruijie(config-if)# ip address 192.168.3.1 255.255.255.0
Ruijie(config-if)# bfd interval 200 min_rx 200 multiplier 5
```

# Configure the Routed Port GE 1/1.

```
Ruijie(config-if)# exit
Ruijie(config)# interface GigabitEthernet1/1
Ruijie(config-if)# no switchport (This configuration is unnecessary for routers)
Ruijie(config)# ip address 192.168.1.1 255.255.255.0
```

# Enable OSPF and configure the BFD for OSPF to detect the neighbor 192.168.3.2.

```
Ruijie(config-if)# exit
Ruijie(config-router)# router ospf 123
Ruijie(config-router)# log-adj-changes detail
Ruijie(config-router)# network 192.168.3.0 0.0.0.255 area 0
Ruijie(config-router)# network 192.168.1.0 0.0.0.255 area 0
Ruijie(config-router)# bfd all-interfaces
Ruijie(config-router)# end
Ruijie#
```

- Router B Configuration

# Configure the Routed Port, the IP address, and the BFD session parameter for Router B.

```
Ruijie# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# interface GigabitEthernet 2/1
Ruijie(config-if)# no switchport (This configuration is unnecessary for routers)
Ruijie(config-if)# ip address 192.168.3.2 255.255.255.0
Ruijie(config-if)# bfd interval 50 min_rx 50 multiplier 3
```

# Configure the Routed Port GE 1/1.

```
Ruijie(config-if)# exit
Ruijie(config)# interface GigabitEthernet1/1
Ruijie(config-if)# no switchport (This configuration is unnecessary for routers)
Ruijie(config-if)# ip address 192.168.2.1 255.255.255.0
```

# Enable OSPF and configure the BFD for OSPF to detect the neighbor 192.168.3.1.

```
Ruijie(config-if)# exit
Ruijie(config-router)# router ospf 123
Ruijie(config-router)# log-adj-changes detail
Ruijie(config-router)# network 192.168.3.0 0.0.0.255 area 0
Ruijie(config-router)# network 192.168.2.0 0.0.0.255 area 0
Ruijie(config-router)# bfd all-interfaces
Ruijie(config-router)# end
Ruijie#
```

Verification

- View the BFD session of Router A

```
Ruijie# show bfd neighbors details
OurAddr         NeighAddr      LD/RD   RH  Holdown(mult)   State   Int
192.168.3.1     192.168.3.2    1/2     1       532 (3 )        Up      Ge2/1
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 200000, MinRxInt: 200000, Multiplier: 5
Received MinRxInt: 50000, Received Multiplier: 3
Holdown (hits): 600(22), Hello (hits): 200(84453)
Rx Count: 49824, Rx Interval (ms) min/max/avg: 208/440/332
Tx Count: 84488, Tx Interval (ms) min/max/avg: 152/248/196
Registered protocols: OSPF
Uptime: 02:18:49
Last packet:    Version: 1             - Diagnostic: 0
I Hear You bit: 1       - Demand bit: 0
Poll bit: 0             - Final bit: 0
Multiplier: 3               - Length: 24
My Discr.: 2            - Your Discr.: 1
Min tx interval: 50000  - Min rx interval: 50000
Min Echo interval: 0
```

- View the BFD session of Router B

```
Ruijie# show bfd neighbors details
OurAddr         NeighAddr       LD/RD   RH  Holdown(mult)   State   Int
192.168.3.2     192.168.3.1     2/1     1       532 (5 )        Up      Ge2/1
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 50000, MinRxInt: 50000, Multiplier: 3
Received MinRxInt: 200000, Received Multiplier: 5
Holdown (hits): 600(22), Hello (hits): 200(84453)
Rx Count: 49824, Rx Interval (ms) min/max/avg: 209/440/332 last: 66 ms ago
Tx Count: 84488, Tx Interval (ms) min/max/avg: 153/249/197 last: 190 ms ago
Registered protocols: OSPF
Uptime: 02:18:49
Last packet:    Version: 1             - Diagnostic: 0
I Hear You bit: 1       - Demand bit: 0
Poll bit: 0             - Final bit: 0
Multiplier: 5               - Length: 24
My Discr.: 1            - Your Discr.: 2
Min tx interval: 200000     - Min rx interval: 200000
Min Echo interval: 0
```

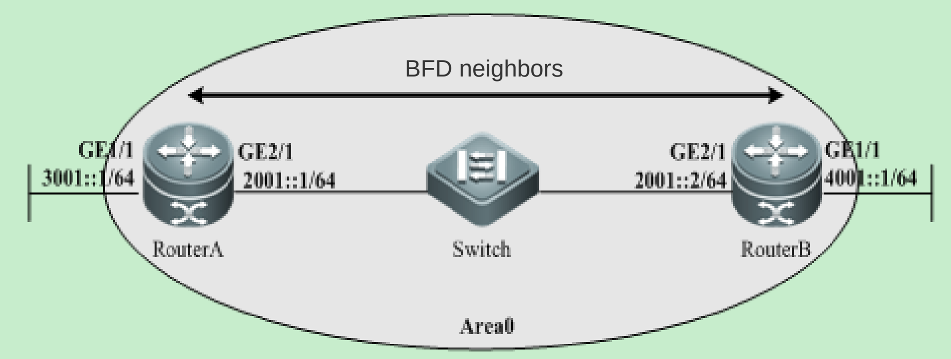
Example of Configuring BFD for OSPFv3

Networking Requirement

Router A and Router B are interconnected through the layer-2 switch. By running the Open Shortest Path First version 3 (OSPFv3) protocol on the switch, routes can be created and association between OSPFv3 and BFD can be enabled on interfaces of both routers. When the link between Router B and the layer-2 switch fails, the BFD function can quickly detect the failure and notify the switch running the OSPFv3 protocol to trigger fast convergence.

Network Topology

Figure 6 Association Between OSPFv3 and BFD



Configuration Tips

- Configure Router A

#Configure an interface (routed port), interface IP address, and BFD session parameters of the interface on Router A.

```
Ruijie# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# interface GigabitEthernet2/1
Ruijie(config-if)# ipv6 ospf 123 area 0
Ruijie(config-if)# ipv6 address 2001::1/64
Ruijie(config-if)# bfd interval 200 min_rx 200 multiplier 5
```

#Configure the interface GE1/1.

```
Ruijie(config-if)# exit
Ruijie(config)# interface GigabitEthernet1/1
Ruijie(config)# ipv6 ospf 123 area 0
Ruijie(config-if)# ipv6 address 3001::1/64
```

#Enable the OSPFv3 protocol and enable the association between OSPFv3 and BFD to detect neighbors of Router B.

```
Ruijie(config-if)# exit
Ruijie(config-router)# router ospf 123
Ruijie(config-router)# log-adj-changes detail
Ruijie(config-router)# bfd all-interfaces
Ruijie(config-router)# end
Ruijie#
```

- Configure Router B

#Configure an interface (routed port), interface IP address, and BFD session parameters of the interface on Router B.

```
Ruijie# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# interface GigabitEthernet 2/1
Ruijie(config-if)# ipv6 ospf 123 area 0
Ruijie(config-if)# ipv6 address 2001::1/64
Ruijie(config-if)# bfd interval 50 min_rx 50 multiplier 3
```

#Configure the interface GE1/1.

```
Ruijie(config-if)# exit
Ruijie(config)# interface GigabitEthernet1/1
Ruijie(config)# ipv6 ospf 123 area 0
Ruijie(config-if)# ipv6 address 4001::1/64
```

#Enable the OSPFv3 protocol and enable the association between OSPFv3 and BFD to detect neighbors of Router A.

```
Ruijie(config-if)# exit
Ruijie(config-router)# ipv6 router ospf 123
Ruijie(config-router)# log-adj-changes detail
Ruijie(config-router)# bfd all-interfaces
Ruijie(config-router)# end
Ruijie#
```

Verification

- Display BFD session establishment information on Router A.

```
Ruijie# show bfd neighbors details
OurAddr         NeighAddr       LD/RD   RH/RS   Holdown(mult)   State   Int
fe80::200:12ff:fe34:5678 fe80::21a:a9ff:fe41:a124  1/2     Up      532 (3 )        Up
    Ge2/1
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 200000, MinRxInt: 200000, Multiplier: 5
Received MinRxInt: 50000, Received Multiplier: 3
Holdown (hits): 600(22), Hello (hits): 200(84453)
Rx Count: 49824, Rx Interval (ms) min/max/avg: 208/440/332
Tx Count: 84488, Tx Interval (ms) min/max/avg: 152/248/196
Registered protocols: OSPFv3
Uptime: 02:18:49
Last packet:    Version: 1             - Diagnostic: 0
I Hear You bit: 1      - Demand bit: 0
Poll bit: 0            - Final bit: 0
Multiplier: 3              - Length: 24
My Discr.: 2           - Your Discr.: 1
Min tx interval: 50000 - Min rx interval: 50000
Min Echo interval: 0
```

- Display BFD session establishment information on Router B.

```
Ruijie# show bfd neighbors details
OurAddr         NeighAddr       LD/RD   RH/RS   Holdown(mult)   State   Int
fe80::200:12ff:fe34:5678 fe80::21a:a9ff:fe41:a124  2/1     Up      532 (5 )        Up
    Ge2/1
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 50000, MinRxInt: 50000, Multiplier: 3
Received MinRxInt: 200000, Received Multiplier: 5
Holdown (hits): 600(22), Hello (hits): 200(84453)
Rx Count: 49824, Rx Interval (ms) min/max/avg: 209/440/332 last: 66 ms ago
Tx Count: 84488, Tx Interval (ms) min/max/avg: 153/249/197 last: 190 ms ago
Registered protocols: OSPFv3
Uptime: 02:18:49
Last packet:    Version: 1             - Diagnostic: 0
I Hear You bit: 1      - Demand bit: 0
Poll bit: 0            - Final bit: 0
Multiplier: 5              - Length: 24
My Discr.: 1           - Your Discr.: 2
```

```
Min tx interval: 200000     - Min rx interval: 200000
Min Echo interval: 0
```

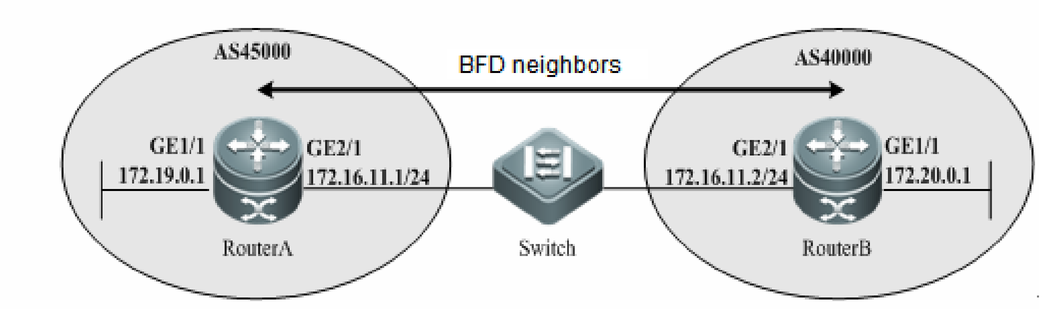
Example of Configuring BFD for BGP

Networking Requirement

Router A and Router B are interconnected through a L2 switch. Both routers run the BGP protocol and enable the BFD for BGP on the interface. After a link failure between Router B and L2 switch occurs, BFD detects the failure and notifies the BGP of the failure, triggering the rapid convergence.

Networking Topology

Figure 7 Topology of configuring BFD for BGP



Configuration Tips

- Router A Configuration

# Configure the Routed Port, the IP address, and the BFD session parameter for Router A.

```
Ruijie# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# interface GigabitEthernet2/1
Ruijie(config-if)# no switchport (This configuration is unnecessary for routers)
Ruijie(config-if)# ip address 172.16.11.1 255.255.255.0
Ruijie(config-if)# bfd interval 200 min_rx 200 multiplier 5
```

# Configure the Routed Port GE 1/1.

```
Ruijie(config-if)# exit
Ruijie(config)# interface GigabitEthernet1/1
Ruijie(config-if)# no switchport (This configuration is unnecessary for routers)
Ruijie(config)# ip address 172.19.0.1 255.255.255.0
```

# Enable BGP and configure the BFD for BGP to detect the neighbor 172.16.11.2.

```
Ruijie(config-if)# exit
Ruijie(config)# router bgp 45000
Ruijie(config-router)# bgp log-neighbor-changes
Ruijie(config-router)# neighbor 172.16.11.2 remote-as 40000
Ruijie(config-router)# neighbor 172.16.11.2 fall-over bfd
Ruijie(config-router)# address-family ipv4
Ruijie(config-router-af)# neighbor 172.16.11.2 activate
Ruijie(config-router-af)# no auto-summary
Ruijie(config-router-af)# no synchronization
Ruijie(config-router-af)# network 172.19.0.0 mask 255.255.255.0
Ruijie(config-router-af)# exit-address-family
Ruijie(config-router)# end
Ruijie#
```

- Router B Configuration

# Configure the Routed Port, the IP address, and the BFD session parameter for Router B.

```
Ruijie# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# interface GigabitEthernet2/1
Ruijie(config-if)# no switchport (This configuration is unnecessary for routers)
Ruijie(config-if)# ip address 172.16.11.2 255.255.255.0
Ruijie(config-if)# bfd interval 50 min_rx 50 multiplier 3
```

# Configure the Routed Port GE 1/1.

```
Ruijie(config-if)# exit
Ruijie(config)# interface GigabitEthernet1/1
Ruijie(config-if)# no switchport (This configuration is unnecessary for routers)
Ruijie(config)# ip address 172.20.0.1 255.255.255.0
```

# Enable BGP and configure the BFD for BGP to detect the neighbor 172.16.11.1.

```
Ruijie(config-if)# exit
Ruijie(config-router)# router bgp 40000
Ruijie(config-router)# bgp log-neighbor-changes
Ruijie(config-router)# neighbor 172.16.11.1 remote-as 45000
Ruijie(config-router)# neighbor 172.16.11.1 fall-over bfd
Ruijie(config-router)# address-family ipv4
Ruijie(config-router-af)# neighbor 172.16.11.1 activate
Ruijie(config-router-af)# no auto-summary
Ruijie(config-router-af)# no synchronization
Ruijie(config-router-af)# network 172.20.0.0 mask 255.255.255.0
Ruijie(config-router-af)# exit-address-family
Ruijie(config-router)# end
Ruijie#
```

Verification

- View the BFD session of Router A.

```
Ruijie# show bfd neighbors details
OurAddr         NeighAddr   LD/RD   RH/RS   Holdown(mult)   State   Int
172.16.11.1     172.16.11.2    1/2     Up      532 (3 )           Up      Ge2/1
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 200000, MinRxInt: 200000, Multiplier: 5
Received MinRxInt: 50000, Received Multiplier: 3
Holdown (hits): 600(22), Hello (hits): 200(84453)
Rx Count: 49824, Rx Interval (ms) min/max/avg: 208/440/332
Tx Count: 84488, Tx Interval (ms) min/max/avg: 152/248/196
Registered protocols: BGP
Uptime: 02:18:49
Last packet: Version: 1    - Diagnostic: 0
I Hear You bit: 1          - Demand bit: 0
Poll bit: 0            - Final bit: 0
Multiplier: 3          - Length: 24
My Discr.: 2               - Your Discr.: 1
Min tx interval: 50000     - Min rx interval: 50000
Min Echo interval: 0
```

- View the BFD session of Router B.

```
Ruijie# show bfd neighbors details
OurAddr         NeighAddr   LD/RD   RH/RS   Holdown(mult)   State   Int
172.16.11.2     172.16.11.1    2/1     Up      532 (5 )           Up      Ge2/1
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 50000, MinRxInt: 50000, Multiplier: 3
Received MinRxInt: 200000, Received Multiplier: 5
Holdown (hits): 600(22), Hello (hits): 200(84453)
Rx Count: 49824, Rx Interval (ms) min/max/avg: 209/440/332 last: 66 ms ago
Tx Count: 84488, Tx Interval (ms) min/max/avg: 153/249/197 last: 190 ms ago
Registered protocols: BGP
Uptime: 02:18:49
Last packet: Version: 1    - Diagnostic: 0
I Hear You bit: 1          - Demand bit: 0
Poll bit: 0            - Final bit: 0
Multiplier: 5          - Length: 24
My Discr.: 1               - Your Discr.: 2
Min tx interval: 200000    - Min rx interval: 200000
Min Echo interval: 0
```

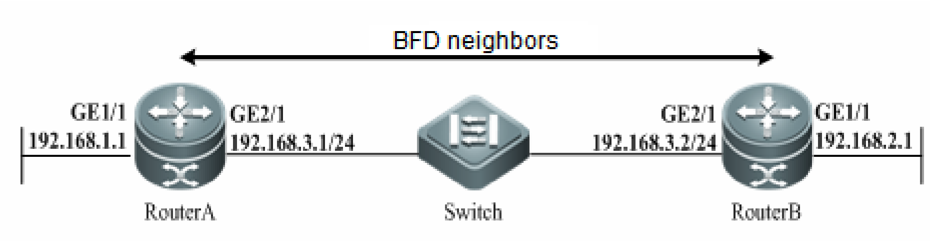
Example of Configuring BFD for Static Route

Networking Requirement

Router A and Router B are interconnected through a L2 switch. Both routers run the static route protocol and enable the BFD for static route on the interface. After a link failure between Router B and L2 switch occurs, BFD detects the failure and notifies the static route of the failure, triggering the static route removal from RIB and preventing the routing error.

Networking Topology

Figure 8 Topology of configuring BFD for Static Route



Configuration Tips

- Router A Configuration

# Configure the Routed Port, the IP address, and the BFD session parameter for Router A.

```
Ruijie# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# interface GigabitEthernet2/1
Ruijie(config-if)# no switchport (This configuration is unnecessary for routers)
Ruijie(config-if)# ip address 192.168.3.1 255.255.255.0
Ruijie(config-if)# bfd interval 200 min_rx 200 multiplier 5
```

# Configure the Routed Port GE 1/1.

```
Ruijie(config-if)# exit
Ruijie(config)# interface GigabitEthernet1/1
Ruijie(config-if)# no switchport (This configuration is unnecessary for routers)
Ruijie(config)# ip address 192.168.1.1 255.255.255.0
```

# Configure the BFD for static route to detect the neighbor 192.168.3.2.

```
Ruijie(config-if)# exit
Ruijie(config)# ip route static bfd GigabitEthernet 2/1 192.168.3.2
Ruijie(config)# ip route 192.168.2.0 255.255.255.0 GigabitEthernet 2/1 192.168.3.2
Ruijie(config)# end
Ruijie#
```

- Router B Configuration

# Configure the Routed Port, the IP address, and the BFD session parameter for Router B.

```
Ruijie# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# interface GigabitEthernet 2/1
Ruijie(config-if)# no switchport (This configuration is unnecessary for routers)
Ruijie(config-if)# ip address 192.168.3.2 255.255.255.0
Ruijie(config-if)# bfd interval 50 min_rx 50 multiplier 3
```

# Configure the Routed Port GE 1/1.

```
Ruijie(config-if)# exit
Ruijie(config)# interface GigabitEthernet1/1
Ruijie(config-if)# no switchport (This configuration is unnecessary for routers)
Ruijie(config-if)# ip address 192.168.2.1 255.255.255.0
```

# Configure the BFD for static route to detect the neighbor 192.168.3.1.

```
Ruijie(config-if)# exit
Ruijie(config)# ip route static bfd GigabitEthernet 2/1 192.168.3.1
Ruijie(config)# ip route 192.168.1.0 255.255.255.0 GigabitEthernet 2/1 192.168.3.1
Ruijie(config)# end
Ruijie#
```

Verification

- View the BFD session of Router A.

```
Ruijie# show bfd neighbors details
OurAddr         NeighAddr       LD/RD   RH  Holdown(mult)   State   Int
192.168.3.1     192.168.3.2     1/2     1       532 (3 )        Up      Ge2/1
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 200000, MinRxInt: 200000, Multiplier: 5
Received MinRxInt: 50000, Received Multiplier: 3
Holdown (hits): 600(22), Hello (hits): 200(84453)
Rx Count: 49824, Rx Interval (ms) min/max/avg: 208/440/332
Tx Count: 84488, Tx Interval (ms) min/max/avg: 152/248/196
Registered protocols: STATIC ROUTE
Uptime: 02:18:49
Last packet:    Version: 1             - Diagnostic: 0
I Hear You bit: 1      - Demand bit: 0
Poll bit: 0            - Final bit: 0
Multiplier: 3              - Length: 24
My Discr.: 2           - Your Discr.: 1
Min tx interval: 50000 - Min rx interval: 50000
Min Echo interval: 0
```

- View the BFD session of Router B.

```
Ruijie# show bfd neighbors details
OurAddr         NeighAddr       LD/RD   RH  Holdown(mult)   State   Int
192.168.3.2     192.168.3.1     2/1     1       532 (5 )        Up      Ge2/1
```

```
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 50000, MinRxInt: 50000, Multiplier: 3
Received MinRxInt: 200000, Received Multiplier: 5
Holdown (hits): 600(22), Hello (hits): 200(84453)
Rx Count: 49824, Rx Interval (ms) min/max/avg: 209/440/332 last: 66 ms ago
Tx Count: 84488, Tx Interval (ms) min/max/avg: 153/249/197 last: 190 ms ago
Registered protocols: STATIC ROUTE
Uptime: 02:18:49
Last packet:    Version: 1             - Diagnostic: 0
I Hear You bit: 1      - Demand bit: 0
Poll bit: 0            - Final bit: 0
Multiplier: 5              - Length: 24
My Discr.: 1           - Your Discr.: 2
Min tx interval: 200000    - Min rx interval: 200000
Min Echo interval: 0
```

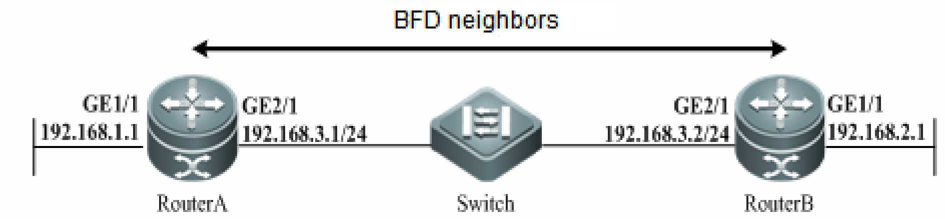
Example of Configuring BFD for PBR

Networking Requirement

Router A and Router B are interconnected through a L2 switch. Both routers run the PBR protocol and enable the BFD for PBR on the interface. After a link failure between Router B and L2 switch occurs, BFD detects the failure and notifies the PBR of the failure, triggering the PBR removal and preventing the routing error.

Networking Topology

Figure 9 Topology of configuring BFD for PBR



Configuration Tips

- Router A Configuration

# Configure the Routed Port GE2/1, the interface IP address, and the BFD session parameter of the interface for Router A.

```
Ruijie# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# interface GigabitEthernet2/1
Ruijie(config-if)# no switchport (This configuration is unnecessary for routers)
Ruijie(config-if)# ip address 192.168.3.1 255.255.255.0
Ruijie(config-if)# bfd interval 200 min_rx 200 multiplier 5
```

# Configure the Routed Port GE 1/1.

```
Ruijie(config-if)# exit
Ruijie(config)# interface GigabitEthernet1/1
Ruijie(config-if)# no switchport (This configuration is unnecessary for routers)
Ruijie(config)# ip address 192.168.1.1 255.255.255.0
```

# Configure the BFD for PBR to detect the neighbor 192.168.3.2.

```
Ruijie(config)# ip access-list extended 100
Ruijie(config-ext-nacl)# permit ip any 192.168.2.0 0.0.0.255
Ruijie(config-ext-nacl)# deny ip any any
Ruijie(config-ext-nacl)# exit
Ruijie(config)# route-map Example1 permit 10
Ruijie(config-route-map)# match ip address 100
Ruijie(config-route-map)# set ip precedence priority
Ruijie(config-route-map)#set ip next-hop verify-availability 192.168.3.2 bfd GigabitEthernet
0/1 192.168.3.2
Ruijie(config)# end
Ruijie#
```

- Router B Configuration

# Configure the Routed Port, the IP address, and the BFD session parameter for Router B.

```
Ruijie# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# interface GigabitEthernet 2/1
Ruijie(config-if)# no switchport (This configuration is unnecessary for routers)
Ruijie(config-if)# ip address 192.168.3.2 255.255.255.0
Ruijie(config-if)# bfd interval 50 min_rx 50 multiplier 3
```

# Configure the Routed Port GE 1/1.

```
Ruijie(config-if)# exit
Ruijie(config)# interface GigabitEthernet1/1
Ruijie(config-if)# no switchport (This configuration is unnecessary for routers)
Ruijie(config-if)# ip address 192.168.2.1 255.255.255.0
```

# Configure the BFD for PBR to detect the neighbor 192.168.3.1.

```
Ruijie(config)# ip access-list extended 100
Ruijie(config-ext-nacl)# permit ip any 192.168.1.0 0.0.0.255
Ruijie(config-ext-nacl)# deny ip any any
Ruijie(config-ext-nacl)# exit
Ruijie(config)# route-map Example1 permit 10
Ruijie(config-route-map)# match ip address 100
Ruijie(config-route-map)# set ip precedence priority
Ruijie(config-route-map)#set ip next-hop verify-availability 192.168.3.1 bfd GigabitEthernet
2/1 192.168.3.1
Ruijie(config)# end
Ruijie#
```

Verification

- View the BFD session of Router A.

```
Ruijie# show bfd neighbors details
OurAddr         NeighAddr      LD/RD   RH  Holdown(mult)  State   Int
192.168.3.1     192.168.3.2    1/2     1       532 (3 )       Up      Ge2/1
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 200000, MinRxInt: 200000, Multiplier: 5
Received MinRxInt: 50000, Received Multiplier: 3
Holdown (hits): 600(22), Hello (hits): 200(84453)
Rx Count: 49824, Rx Interval (ms) min/max/avg: 208/440/332
Tx Count: 84488, Tx Interval (ms) min/max/avg: 152/248/196
Registered protocols: PBR
Uptime: 02:18:49
Last packet:    Version: 1             - Diagnostic: 0
I Hear You bit: 1      - Demand bit: 0
Poll bit: 0            - Final bit: 0
Multiplier: 3              - Length: 24
My Discr.: 2           - Your Discr.: 1
Min tx interval: 50000 - Min rx interval: 50000
Min Echo interval: 0
```

- View the BFD session of Router B.

```
Ruijie# show bfd neighbors details
OurAddr         NeighAddr      LD/RD   RH  Holdown(mult)  State   Int
192.168.3.2     192.168.3.1    2/1     1       532 (5 )       Up      Ge2/1
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 50000, MinRxInt: 50000, Multiplier: 3
Received MinRxInt: 200000, Received Multiplier: 5
Holdown (hits): 600(22), Hello (hits): 200(84453)
Rx Count: 49824, Rx Interval (ms) min/max/avg: 209/440/332 last: 66 ms ago
Tx Count: 84488, Tx Interval (ms) min/max/avg: 153/249/197 last: 190 ms ago
Registered protocols: PBR
Uptime: 02:18:49
Last packet:    Version: 1             - Diagnostic: 0
I Hear You bit: 1      - Demand bit: 0
Poll bit: 0            - Final bit: 0
Multiplier: 5              - Length: 24
My Discr.: 1           - Your Discr.: 2
Min tx interval: 200000    - Min rx interval: 200000
Min Echo interval: 0
```

Example of Configuring BFD for VRRP

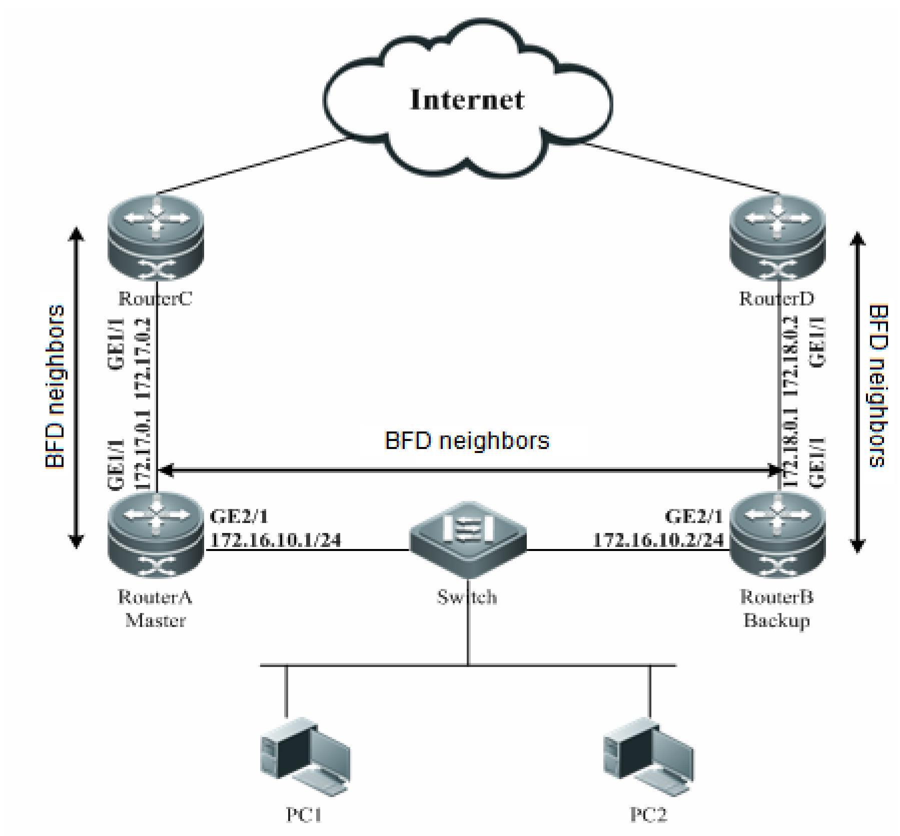
Networking Requirement

Router A and Router B are interconnected through a L2 switch. Both routers run the VRRP protocol and enable the BFD for PBR on the interface to detect the master and backup routers. After a link failure between Router A and L2 switch occurs, BFD detects the failure, notifies VRRP of the failure, and triggers the decrease of the priority of the VRRP master router. As a result, the switchover between the master and backup routers, which enables the backup router rapidly.

Router A and Router B access the Internet through Router C and Router D respectively. Configure the static routes to establish the forwarding path between Router A and Router C, Router B and Router D and enable the BFD to detect the neighbor. At the same time, Router A and Router B are configured the BFD for VRRP to detect the forwarding path between the Router A and Router C, Router B and Router D. The detection failure triggers the decrease of the priority for VRRP master router and switchover between the master and backup routers, which enables the backup router rapidly.

Networking Topology

Figure 10 Topology of configuring BFD for VRRP



Configuration Tips

- Router A Configuration

# Configure the Routed Port, the IP address, and the BFD session parameter for Router A.

```
Ruijie# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# interface GigabitEthernet2/1
Ruijie(config-if)# no switchport (This configuration is unnecessary for routers)
Ruijie(config-if)# ip address 172.16.10.1 255.255.255.0
Ruijie(config-if)# bfd interval 200 min_rx 200 multiplier 5
```

# Configure the Routed Port GE 1/1.

```
Ruijie(config-if)# exit
Ruijie(config)# interface GigabitEthernet1/1
Ruijie(config-if)# no switchport (This configuration is unnecessary for routers)
Ruijie(config-if)# ip address 172.17.0.1 255.255.255.0
Ruijie(config-if)# bfd interval 200 min_rx 200 multiplier 5
```

# Enable VRRP and configure the BFD for VRRP to detect the neighbor 172.16.10.2 and 172.17.0.2 at the same time.

```
Ruijie(config-if)# interface GigabitEthernet2/1
Ruijie(config-if)# vrrp 1 timers advertise 3
Ruijie(config-if)# vrrp 1 ip 172.16.10.3
Ruijie(config-if)# vrrp 1 priority 120
Ruijie(config-if)# vrrp 1 bfd 172.16.10.2
Ruijie(config-if)# vrrp 1 track bfd GigabitEthernet 1/1 172.17.0.2 30
```

# Configure the static route and associate the BFD to detect the neighbor 172.17.0.2:

```
Ruijie(config-if)# exit
Ruijie(config)# ip route static bfd GigabitEthernet 1/1 172.17.0.2
Ruijie(config)# ip route 0.0.0.0 0.0.0.0 GigabitEthernet 1/1 172.17.0.2
Ruijie(config)# end
Ruijie#
```

- Router B Configuration

# Configure the Routed Port, the IP address, and the BFD session parameter for Router B.

```
Ruijie# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# interface GigabitEthernet2/1
Ruijie(config-if)# no switchport (This configuration is unnecessary for routers)
Ruijie(config-if)# ip address 172.16.10.2 255.255.255.0
Ruijie(config-if)# bfd interval 50 min_rx 50 multiplier 3
```

# Configure the Routed Port GE 1/1.

```
Ruijie(config-if)# exit
Ruijie(config)# interface GigabitEthernet1/1
Ruijie(config-if)# no switchport (This configuration is unnecessary for routers)
Ruijie(config-if)# ip address 172.18.0.1 255.255.255.0
Ruijie(config-if)# bfd interval 200 min_rx 200 multiplier 5
```

# Enable VRRP and configure the BFD for VRRP to detect the neighbor 172.16.10.1 and 172.18.0.2 at the same time.

```
Ruijie(config-if)# interface GigabitEthernet2/1
Ruijie(config-if)# vrrp 1 timers advertise 3
Ruijie(config-if)# vrrp 1 ip 172.16.10.3
Ruijie(config-if)# vrrp 1 priority 120
Ruijie(config-if)# vrrp 1 bfd 172.16.10.1
Ruijie(config-if)# vrrp 1 track bfd GigabitEthernet 1/1 172.18.0.2 30
```

# Configure the static route and associate the BFD to detect the neighbor 172.18.0.2.

```
Ruijie(config-if)# exit
Ruijie(config)# ip route static bfd GigabitEthernet 1/1 172.18.0.2
Ruijie(config)# ip route 0.0.0.0 0.0.0.0 GigabitEthernet 1/1 172.18.0.2
Ruijie(config)# end
Ruijie#
```

Verification

- View the BFD session of Router A.

```
Ruijie# show bfd neighbors details
OurAddr         NeighAddr      LD/RD   RH  Holdown(mult)   State   Int
172.16.10.1     172.16.10.2    1/2     1       532 (3 )       Up      Ge2/1
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 200000, MinRxInt: 200000, Multiplier: 5
Received MinRxInt: 50000, Received Multiplier: 3
Holdown (hits): 600(22), Hello (hits): 200(84453)
Rx Count: 49824, Rx Interval (ms) min/max/avg: 208/440/332
Tx Count: 84488, Tx Interval (ms) min/max/avg: 152/248/196
Registered protocols: VRRP
Uptime: 02:18:49
Last packet:    Version: 1             - Diagnostic: 0
I Hear You bit: 1      - Demand bit: 0
Poll bit: 0            - Final bit: 0
Multiplier: 3              - Length: 24
My Discr.: 2           - Your Discr.: 1
Min tx interval: 50000 - Min rx interval: 50000
Min Echo interval: 0
```

```
OurAddr         NeighAddr       LD/RD   RH  Holdown(mult)   State   Int
172.17.0.1  172.17.0.2         2/3     1       532 (3 )       Up      Ge2/1
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 200000, MinRxInt: 200000, Multiplier: 5
Received MinRxInt: 50000, Received Multiplier: 3
Holdown (hits): 600(22), Hello (hits): 200(84453)
Rx Count: 49824, Rx Interval (ms) min/max/avg: 208/440/332 last: 68 ms ago
Tx Count: 84488, Tx Interval (ms) min/max/avg: 152/248/196 last: 192 ms ago
Registered protocols: VRRP,STATIC ROUTE
Uptime: 02:18:49
Last packet:    Version: 1             - Diagnostic: 0
I Hear You bit: 1      - Demand bit: 0
Poll bit: 0            - Final bit: 0
Multiplier: 3              - Length: 24
My Discr.: 2           - Your Discr.: 1
Min tx interval: 50000 - Min rx interval: 50000
Min Echo interval: 0
```

View the BFD session of Router B.

```
Ruijie# show bfd neighbors details
OurAddr         NeighAddr       LD/RD   RH  Holdown(mult)   State   Int
172.16.10.2     172.16.10.1     2/1     1       532 (3 )       Up      Ge2/1
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 50000, MinRxInt: 50000, Multiplier: 3
Received MinRxInt: 200000, Received Multiplier: 5
Holdown (hits): 600(22), Hello (hits): 200(84453)
Rx Count: 49824, Rx Interval (ms) min/max/avg: 208/440/332 last: 68 ms ago
Tx Count: 84488, Tx Interval (ms) min/max/avg: 152/248/196 last: 192 ms ago
Registered protocols: VRRP
Uptime: 02:18:49
Last packet:    Version: 1             - Diagnostic: 0
I Hear You bit: 1      - Demand bit: 0
Poll bit: 0            - Final bit: 0
Multiplier: 3              - Length: 24
My Discr.: 1           - Your Discr.: 2
Min tx interval: 200000    - Min rx interval: 200000
Min Echo interval: 0

OurAddr         NeighAddr       LD/RD   RH  Holdown(mult)   State   Int
172.18.0.1      172.18.0.2      1/3     1       532 (3 )       Up      Ge2/1
Local Diag: 0, Demand mode: 0, Poll bit: 0
MinTxInt: 200000, MinRxInt: 200000, Multiplier: 5
Received MinRxInt: 50000, Received Multiplier: 3
Holdown (hits): 600(22), Hello (hits): 200(84453)
Rx Count: 49824, Rx Interval (ms) min/max/avg: 208/440/332 last: 68 ms ago
```

```
Tx Count: 84488, Tx Interval (ms) min/max/avg: 152/248/196 last: 192 ms ago
Registered protocols: VRRP,STATIC ROUTE
Uptime: 02:18:49
Last packet:    Version: 1              - Diagnostic: 0
I Hear You bit: 1       - Demand bit: 0
Poll bit: 0             - Final bit: 0
Multiplier: 3               - Length: 24
My Discr.: 2            - Your Discr.: 1
Min tx interval: 50000 - Min rx interval: 50000
Min Echo interval: 0
```

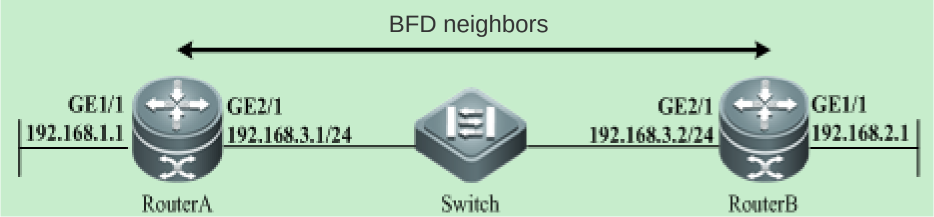
Example of Configuring QoS Policy in ECHO Mode for Guaranteeing Priority

Networking Requirement

Router A and Router B are interconnected through the layer-2 switch. A traffic pumping device of IXIA is used to pump traffic from Router A to Router B. A QoS policy is configured on Router A to guarantee the forwarding priority of echo packets sent from Router B.

Network Topology

Figure 11 QoS Policy Configuration in ECHO Mode



Configuration Tips

- Configuring Router A

#Configure packets of Port 3785 as BFD echo packets in the ACL.

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)#ip access-list extended 100
Ruijie(config-ext-nacl)#10 permit udp any any eq 3785
Ruijie(config-ext-nacl)#
```

#Configure the port queue template.

```
Ruijie(config)#port-queue 1
```

#Configure the user queue.

```
Ruijie(config)#user-queue uq1 outbound
Ruijie(config-user-queue)# cir 1000000 pir 1000000
Ruijie(config-user-queue)#
Ruijie#
```

#Configure the traffic classifier.

```
Ruijie(config)#traffic classifier tc1 or
Ruijie(config-traffic-classifier)# if-match acl 100
Ruijie(config-traffic-classifier)#
Ruijie#
```

#Configure the traffic behavior.

```
Ruijie(config)#traffic behavior tb1
Ruijie(config-traffic-behavior)# user-queue uq1 outbound
Ruijie(config-traffic-behavior)# service-class cs7 color green
Ruijie(config-traffic-behavior)#
```

#Configure the traffic policy.

```
Ruijie(config)#traffic policy tp1
Ruijie(config-traffic-policy)# classifier tcl behavior tb1 precedence 1
Ruijie(config-traffic-policy)#
```

#Apply the policy to traffic from the interface to application.

```
Ruijie(config)# interface GigabitEthernet2/1
Ruijie(config-if- GigabitEthernet2/1)#traffic-policy tp1 outbound
Ruijie(config-if- GigabitEthernet2/1)#
```

Example of Configuring BFD for the L3 Interface

Because the configuration of BFD for the L3 interface is usually used in FRR application and independent usage is not recommended. For details, see description in *MPLS-SCG.doc*.

Example of Configuring BFD for MPLS

For details, see description in *MPLS-SCG.doc*.

Example of Configuring BFD for VRRP

For details, see description in *VRRP-PLUS-SCG.doc*.

Dialing Configuration

1 Configuring WAN-4G

1.1 Overview

1.1.1 Brief Introduction

The fourth generation (4G) of mobile communication technology is a kind of new mobile communication technology. with a theoretical maximum data transmission rate of 100 Mbps, 4G network can support such high-bandwidth applications as HD video transmission and satellite communication. The following gives a description of some basic concepts involved in 4G function configuration.

1.1.2 Basic Concepts

APN

The Access Point Name (APN) determines the access mode through which the user accesses the 4G network. Currently, Ruijie devices support configuration of 3GNET.

PCO

User name and password.

Track

A Track object can track whether an IP address is reachable and whether an interface is up. The Track function separates the object to be tracked from the module that is interested in the status of this object. When the status of the Track object changes, track function modules can adopt different actions.

1.1.3 Working Principle

The network structure of the 4G mobile system can be divided into three layers: physical network layer, intermediate environment layer and application network layer. The physical network layer provides the access and routing functions, which are completed by the combine format of the wireless network and core network. The intermediate environment layer provides the function such as Qu’s mapping, address translation and completeness management. The interface between the physical network layer and the intermediate environment layer and its application environment is open. It facilitates development and provision of new applications and services, provides seamless and high-rate wireless services, and runs on multiple bands.

The configuration commands described here can implement the 4G interface communication functions of Ruijie devices.

1.1.4 Applications

4G integrates 3G and WLAN and can transmit high-quality video images. Its image transmission quality is equivalent to that of high-definition TV. The 4G system supports a download speed of 100 Mbps, a speed 2,000 times faster than that of current dial-up Internet access, and an upload speed of 20 Mbps, which can satisfy the requirements of almost all the users for wireless services. In addition, 4G network can be deployed in places not covered by DSL and cable TV modems and then expanded to the whole region. Obviously, 4G has incomparable superiority.

1.2 Configuring WAN-4G

To configure WAN-4G, you need to configure the following aspects: dialing function, 4G network access mode selection, protocol status of associated master interface, dial-on-demand, and anti-impact.

1.2.1 Single Card and Single Access Dialing Function

1.2.1.1 Configuration steps

Step	Configuration Task	Description
1	Configuring APN and PCO	Optional
2	Configuring APN and PCO	Optional
3	Configuring a Single 4G Interface to Associate with Track	Mandatory

1.2.1.2 Configuring APN and PCO

➢ Configuration method 1:

Command	Description
Ruijie# configure terminal	Enters the global configuration mode.
Ruijie(config)# interface cellular *0/0*	Enters the interface configuration mode.
Ruijie(config-if-Cellular1)# profile create master apn *3gnet*	Configures APN.
Ruijie(config-if-Cellular 0/0)# profile create master username 3gnet password 0 3gnet	Configures the specified user name/password.

1: A public network SIM card can access the network normally without configuring the APN. The APN can also be configured manually by referring to the public network access requirements of the operator's network.

2: The APN needs to be correctly configured for a dedicated line SIM card. If there is an APN configuration error, the SIM card may access a public network.

3: After the APN or PCO configuration is modified, validate the configuration by using any one of the following three methods: Method 1: Execute **reset** on the interface to reset the interface. Method 2: Execute **shutdown** on the interface, and then execute **no shutdown** after waiting for at least two seconds. Method 3: Save the configuration and restart the host. If the configuration under the 4G interface is modified by remotely logging in to the device through the 4G line, only method 1 or 2 can be used. Any of the three methods can be used if the configuration is not modified through the 4G line.

Example: Configure the APN and PCO functions on the interface cellular 0/0.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#interface cellular 0/0
Ruijie(config-if-Cllular 0/0)#profile create master apn 3gnet
Ruijie(config-if-Cllular 0/0)#profile create master username user1 password pass1
Ruijie(config-if-Cllular 0/0)#show run interface cellular 0/0

Building configuration...
```

```
Current configuration: 141 bytes
!
interface cellular 0/0
 profile create master apn 3gnet
profile create master username user1 password pass1
```

Configuration method 2 (recommended):

Command	Description
Ruijie# configure terminal	Enters the global configuration mode.
Ruijie(config)# interface cellular *0/0*	Enters the interface configuration mode.
Ruijie(config-if-Cellular 0/0)# plmn sent-username *abc* password 0 *123* [apn *3gnet*]	Configures the user name, password and APN (optional).

1: A public network SIM card can access the network normally without configuring the APN. The APN can also be configured manually by referring to the public network access requirements of the operator's network.

2: The APN needs to be correctly configured for a dedicated line SIM card. If there is an APN configuration error, the SIM card may access a public network.

3: The dialing time is usually within 20s when the mobile network signal is normal. The dialing time is relatively long when the signal is poor. Dialing attempts will continue until the dialing succeeds. The modem will be reset during the period, and then dialing attempts are continuously made.

4: After the router is restarted or the modem is reset, the modem needs to be initialized. In this case, the dialing time is about 90s.

Example: Configure the PCO and APN functions on the interface cellular 0/0.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#interface cellular 0/0
Ruijie(config-if-Cllular 0/0)# plmn sent-username abc password 0 123 apn 3gnet
Ruijie(config-if-Cllular 0/0)#show run interface cellular 0/0

Building configuration...
Current configuration : 141 bytes
!
interface cellular 0/0
 plmn sent-username abc password 0 123 apn 3gnet
```

1.2.1.3 Configuring a Single 4G Interface to Associate with Track

➢ Configuration method 1:

Command	Description
Ruijie# configure terminal	Enters the global configuration mode.
Ruijie(config)# ip rns *1*	Enters the IP RNS configuration mode.
Ruijie(config-ip-rns)# icmp-echo *10.1.1.1*	Configures an IP RNS object and uses it to send ICMP packets.
Ruijie(config-ip-rns)# frequency *1000*	The interval for the RNS to send packets, in milliseconds. This interval must be greater than or equal to the timeout duration. The value is 60 seconds by default. The value range is from 10 to 604800000.
Ruijie(config-ip-rns)# timeout *1000*	The time used to judge timeout after the RNS sends packets, in milliseconds. The value is 5 seconds for the ICMP Echo packet and 9 seconds for the DNS packet by default. The value range is from 10 to 604800000 when ICMP Echo detection is used and from 1000 to 604800000 when DNS detection is used.
Ruijie(config)# track *1* rns *1*	Tracks the status of an IP RNS object and enters the Track mode.
Ruijie(config-track)# delay up *30*	(Optional) Specifies a period of time. When the status of a Track object changes, the status will change after this period of time. There is no delay by default.
Ruijie# configure terminal	Enters the global configuration mode.
Ruijie(config)# interface cellular *0/0*	Enters the interface configuration mode.
Ruijie(config-if- cellular 1)# profile create master track *1*	Configures the ID of the Track object associated with the 4G interface status.
Ruijie(config-if- cellular 1)# backup-valid-check valid-timer 30 max-check-times 3	Configures the time and count of checking the availability query timer (60 seconds and 3 times by default).

Association of a 4G interface with Track can be configured on a Ruijie 4G router to check the link service status. When the status of the Track object changes to down, 4G link reset can be triggered to initiate a dial-up connection. After successful dialing, if the Track status is not up and remains to be down, the corresponding 4G interface is still unavailable in this situation. In this case, this timer needs to be started to perform check. If it is within the time of set "seconds * max-times", the time is 60s * 3 = 180s (the default configuration is used as an example), the corresponding Track status is still down, and the 4G link is reset again to initiate a dial-up connection, thus ensuring link recovery.

Example: Configure a single 4G card to associate with Track.

```
Ruijie# configure terminal
Enter configuration commands, one per line.   End with CNTL/Z.
Ruijie(config)# ip rns 1
Ruijie(config-ip-rns)# icmp-echo 10.1.1.1
Ruijie(config-ip-rns)# frequency 1000
Ruijie(config-ip-rns)# timeout 1000
Ruijie(config)# track 2 rns 1
Ruijie(config-track)# delay up 30
Ruijie(config-track)# exit
Ruijie(config)# interface cellular 0/0
Ruijie(config-if-cellular 0/0)# profile create master track 1
Ruijie(config-if-cellular 0/0)# backup-valid-check valid-timer 30 max-check-times 3
Ruijie(config-if-cellular 0/0)# show run interface cellular 0/0
Building configuration...
Current configuration : 141 bytes
!
interface cellular 0/0
 backup-valid-check valid-timer 30 max-check-times 3
 profile create master apn 3gnet
 profile create master username 3gnet password 3gnet
 profile create master track 1
```

➢ Configuration method 2 (recommended):

Command	Description
Ruijie# configure terminal	Enters the global configuration mode.
Ruijie(config)# ip rns *1*	Enters the IP RNS configuration mode.
Ruijie(config-ip-rns)# icmp-echo *10.1.1.1*	Configures an IP RNS object and uses it to send ICMP packets.
Ruijie(config-ip-rns)# frequency *1000*	The interval for the RNS to send packets, in milliseconds. This interval must be greater than or equal to the timeout duration. The value is 60 seconds by default. The value range is from 10 to 604800000.
Ruijie(config-ip-rns)# timeout *1000*	The time used to judge timeout after the RNS sends packets, in milliseconds. The value is 5 seconds for the ICMP Echo packet and 9 seconds for the DNS packet by default. The value range is from 10 to 604800000 when ICMP Echo detection is used and from 1000 to 604800000 when DNS detection is used.
Ruijie(config)# track *1* rns *1*	Tracks the status of an IP RNS object and enters the Track mode.
Ruijie(config-track)# delay up *30*	(Optional) Specifies a period of time. When the status of a Track object changes, the status will change after this period of time. There is no delay by default.
Ruijie# configure terminal	Enters the global configuration mode.
Ruijie(config)# interface cellular *0/0*	Enters the interface configuration mode.
Ruijie(config-if- cellular 0/0)# plmn status track *1*	Configures the ID of the track object associated with the 4G interface status.

Ruijie(config-if- cellular0/0)#backup-valid-check valid-timer 30 max-check-times 3	Configures the time and count of checking the availability query timer (60 seconds and 3 times by default).

Association of a 4G interface with Track can be configured on a Ruijie 4G router to check the link service status. When the status of the Track object changes to down, 4G link reset can be triggered to initiate a dial-up connection. After successful dialing, if the Track status is not up and remains to be down, the corresponding 4G interface is still unavailable in this situation. In this case, this timer needs to be started to perform check. If it is within the time of set "seconds * max-times", the time is 60s * 3 = 180s (the default configuration is used as an example), the corresponding Track status is still down, and the 4G link is reset again to initiate a dial-up connection, thus ensuring link recovery.

Example: Configure a single 4G card to associate with Track.

```
Ruijie# configure terminal
Enter configuration commands, one per line.    End with CNTL/Z.
Ruijie(config)# ip rns 1
Ruijie(config-ip-rns)# icmp-echo 10.1.1.1
Ruijie(config-ip-rns)# frequency 1000
Ruijie(config-ip-rns)# timeout 1000
Ruijie(config)# track 2 rns 1
Ruijie(config-track)# delay up 30
Ruijie(config-track)# exit
Ruijie(config)# interface cellular 0/0
Ruijie(config-if-cellular 0/0)# plmn status track 1
Ruijie(config-if-cellular 0/0)# backup-valid-check valid-timer 30 max-check-times 3
Ruijie(config-if-cellular 0/0)# show run interface cellular 0/0
Building configuration...
Current configuration : 141 bytes
!
interface cellular 0/0
backup-valid-check valid-timer 30 max-check-times 3
plmn status track 1
```

1.2.1.4 Configuring a Single 4G Interface to Associate with BFD

Command	Description
Ruijie# configure terminal	Enters the global configuration mode.
Ruijie(config)# interface cellular 0/0	Enters the interface configuration mode.
Ruijie(config-if- cellular 0/0)# plmn status bfd	Configures the 4G interface status to associate with BFD.
Ruijie(config-if- cellular0/0)#backup-valid-check valid-timer 30 max-check-times 3	Configures the time and count of querying the availability query timer (60 seconds and 3 times by default).

Association of a 4G interface with BFD can be configured on a Ruijie 4G router to check the link service status. When the status of the BFD object changes to down, 4G link reset can be triggered to initiate a dial-up connection. After successful dialing, if the BFD status is not up and remains to be down, the corresponding 4G interface is still unavailable in this situation. In this case, this timer needs to be started to perform check. If it is within the time of set "seconds * max-times", the time is 60s * 3 = 180s (the default configuration is used as an example), the corresponding Track status is still down, and the 4G link is reset again to initiate a dial-up connection, thus ensuring link recovery.

Example: Configure a single 4G card to associate with BFD.

```
Ruijie# configure terminal
Enter configuration commands, one per line.    End with CNTL/Z.
Ruijie(config)# interface cellular 0/0
Ruijie(config-if-cellular 0/0)# plmn status bfd
Ruijie(config-if-cellular 0/0)# backup-valid-check valid-timer 30 max-check-times 3
Ruijie(config-if-cellular 0/0)# show run interface cellular 0/0
Building configuration...
Current configuration : 141 bytes
!
interface cellular 0/0
 backup-valid-check valid-timer 30 max-check-times 3
 plmn status bfd
```

1.2.1.5 Configuring a Single 4G Interface and RSSI Status Detection

Command	Description
Ruijie# configure terminal	Enters the global configuration mode.
Ruijie(config)# interface cellular 0/0	Enters the interface configuration mode.
Ruijie(config-if- cellular 0/0)# plmn status rssi-detect -100 interval 16 ntimes 4 percent 50	Configures RSSI detection on a 4G interface.

🕮 RSSI status detection can be configured on a Ruijie 4G router to check the link service status for ntimes within the interval. If the current signal intensity is lower than the configured signal intensity value for ntimes * percent, the link is disconnected. Counter check is continued. If the checked signal intensity is greater than the configured signal intensity for ntimes * percent within the interval, the 4G link will be reset again to initiate a dial-up connection, thus ensuring link recovery.

Example: Configure a single 4G card to implement signal intensity detection.

```
Ruijie# configure terminal
Enter configuration commands, one per line.    End with CNTL/Z.
Ruijie(config)# interface cellular 0/0
Ruijie(config-if-cellular 0/0)# plmn status rssi-detect -100 interval 16 ntimes 4 percent 50
Ruijie(config-if-cellular 0/0)# show run interface cellular 0/0
Building configuration...
Current configuration : 141 bytes
!
interface cellular 0/0
 plmn status rssi-detect -100 interval 16 ntimes 4 percent 50
```

1.2.2 Single Card and Multiple Access Dialing Function

This access mode is applicable to the following scenarios: A single card configured with multiple APNs, user names and passwords. Different APNs access the corresponding dedicated line LNS servers. When the LNS server of a dedicated line fails, a normal LNS server can be accessed through another APN.

1.2.2.1 Configuring PCO and APN

Command	Description
Ruijie# configure terminal	Enters the global configuration mode.
Ruijie(config)# interface cellular 0/0	Enters the interface configuration mode.
Ruijie(config-if-Cellular 0/0)# profile creat {master | slave} [apn 3gnet] username abc password {0 | 7} 333 {bfd | track 10 }	Configures the APN (optional), user name and password; track and bfd may not be used when slave is configured.

1: A public network SIM card can access the network normally without configuring the APN. The APN can also be configured manually by referring to the public network access requirements of the operator's network.

2: The APN needs to be correctly configured for a dedicated line SIM card. If there is an APN configuration error, the SIM card may access a public network.

3: Any change in the APN, user name and password related to master arising from this command will trigger redialing. Its effect is equal to one time of shutdown and no shutdown. Changes to the related parameters of slave will trigger redialing only when the system operates under a slave user currently.

Example 1: Configure the PCO and APN functions on the interface cellular 0/0.

```
Ruijie#configure terminal
Enter configuration commands, one per line.    End with CNTL/Z.
Ruijie(config)#interface cellular 0/0
Ruijie(config-if-Cllular 0/0)# profile creat master apn 3gnet username abc password 0 333 track 10
Ruijie(config-if-Cllular 0/0)#show run interface cellular 0/0

Building configuration...
Current configuration : 141 bytes
!
interface cellular 0/0
 profile creat master apn 3gnet username abc password 0 333 track 10
```

1.2.2.2 Configuring a Single 4G Interface to Associate with Track

Command	Description
Ruijie# configure terminal	Enters the global configuration mode.
Ruijie(config)# ip rns 1	Enters the IP RNS configuration mode.
Ruijie(config-ip-rns)# icmp-echo 10.1.1.1	Configures an IP RNS object and uses it to send ICMP packets.

Ruijie(config-ip-rns)# frequency 1000	The interval for the RNS to send packets, in milliseconds. This interval must be greater than or equal to the timeout duration. The value is 60 seconds by default. The value range is from 10 to 604800000.
Ruijie(config-ip-rns)# timeout 1000	The time used to judge timeout after the RNS sends packets, in milliseconds. The value is 5 seconds for the ICMP Echo packet and 9 seconds for the DNS packet by default. The value range is from 10 to 604800000 when ICMP Echo detection is used and from 1000 to 604800000 when DNS detection is used.
Ruijie(config)# track 1 rns 1	Tracks the status of an IP RNS object and enters the Track mode.
Ruijie(config-track)# delay up 30	(Optional) Specifies a period of time. When the status of a Track object changes, the status will change after this period of time. There is no delay by default.
Ruijie# configure terminal	Enters the global configuration mode.
Ruijie(config)# interface cellular 0/0	Enters the interface configuration mode.
Ruijie(config-if- cellular 0/0)# profile creat {master | slave} [apn 3gnet] username abc password {0 | 7} 333 track 10	Configures the ID of the Track object associated with the 4G interface status.
Ruijie(config-if- cellular0/0)#backup-valid-check valid-timer 30 max-check-times 3	Configures the time and count of checking the availability query timer (60 seconds and 3 times by default).

Association of a 4G interface with Track can be configured on a Ruijie 4G router to check the link service status. When the status of the Track object changes to down, 4G link reset can be triggered to initiate a dial-up connection. After successful dialing, if the Track status is not up and remains to be down, the corresponding 4G interface is still unavailable in this situation. In this case, this timer needs to be started to perform check. If it is within the time of set "seconds * max-times", the time is 60s * 3 = 180s (the default configuration is used as an example), the corresponding Track status is still down, and the 4G link is reset again to initiate a dial-up connection, thus ensuring link recovery.

Example: Configure a single 4G card to associate with Track.

```
Ruijie# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)# ip rns 1
Ruijie(config-ip-rns)# icmp-echo 10.1.1.1
Ruijie(config-ip-rns)# frequency 1000
Ruijie(config-ip-rns)# timeout 1000
Ruijie(config)# track 2 rns 1
Ruijie(config-track)# delay up 30
Ruijie(config-track)# exit
Ruijie(config)# interface cellular 0/0
Ruijie(config-if-cellular 0/0)# profile creat master apn 3gnet username abc password 0 333 track 10
Ruijie(config-if-cellular 0/0)# backup-valid-check valid-timer 30 max-check-times 3
Ruijie(config-if-cellular 0/0)# show run interface cellular 0/0
Building configuration...
```

```
Current configuration : 141 bytes
!
interface cellular 0/0
 backup-valid-check valid-timer 30 max-check-times 3
 profile creat master apn 3gnet username abc password 0 333 track 10
```

1.2.2.3 Configuring a Single 4G Interface to Associate with BFD

Command	**Description**
Ruijie# configure terminal	Enters the global configuration mode.
Ruijie(config)# interface cellular 0/0	Enters the interface configuration mode.
Ruijie(config-if- cellular 0/0)# profile creat {master | slave} [apn 3gnet] username abc password {0 | 7} 333 bfd	Configures the 4G interface status to associate with BFD.
Ruijie(config-if- cellular0/0)#backup-valid-check valid-timer 30 max-check-times 3	Configures the time and count of querying the availability query timer (60 seconds and 3 times by default).

Association of a 4G interface with BFD can be configured on a Ruijie 4G router to check the link service status. When the status of the BFD object changes to down, 4G link reset can be triggered to initiate a dial-up connection. After successful dialing, if the BFD status is not up and remains to be down, the corresponding 4G interface is still unavailable in this situation. In this case, this timer needs to be started to perform check. If it is within the time of set "seconds * max-times", the time is 60s * 3 = 180s (the default configuration is used as an example), the corresponding Track status is still down, and the 4G link is reset again to initiate a dial-up connection, thus ensuring link recovery.

Example: Configure a single 4G card to associate with BFD.

```
Ruijie# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)# interface cellular 0/0
Ruijie(config-if-cellular 0/0)# plmn status bfd
Ruijie(config-if-cellular 0/0)# backup-valid-check valid-timer 30 max-check-times 3
Ruijie(config-if-cellular 0/0)# show run interface cellular 0/0
Building configuration...
Current configuration : 141 bytes
!
interface cellular 0/0
 backup-valid-check valid-timer 30 max-check-times 3
 profile creat master apn 3gnet username abc password 0 333 bfd
```

1.2.2.4 Configuring Switching of Different Access Points

Command	Description
Ruijie# configure terminal	Enters the global configuration mode.
Ruijie(config)# interface cellular 0/0	Enters the interface configuration mode.
Ruijie(config-if-Cellular 0/0)# profile switch access-point	Switches the access point information.

1: Backup is configured. This command can be used to switch between the master account and slave account.

2: If only one access point is configured, the device automatically returns to the original access point if switching fails.

Example 1: Configure different access points on the interface cellular 0/0.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#interface cellular 0/0
Ruijie(config-if-Cllular 0/0)# profile switch access-point
Ruijie(config-if-Cllular 0/0)#show run interface cellular 0/0

Building configuration...
Current configuration : 141 bytes
!
interface cellular 0/0
```

1.2.3 Selecting 4G Network Access Mode

1.2.3.1 Configuration steps

Step	Configuration Task	Description
1	Configuring to Directly Set Current Access Mode (manual handover between 3G/4G)	Mandatory

1.2.3.2 Configuring to Directly Set Current Access Mode

Command	Description
Ruijie# configure terminal	Enters the global configuration mode.
Ruijie(config)# interface cellular *0/0*	Enters the interface configuration mode.
Ruijie(config-if-Cellular 1)# plmn mode manual **{ lte | wcdma | ehrpd | gsm | td-scdma | lte-evdo-cdma }**	Configures the current access mode: **lte, wcdma, ehrpd, gsm, td-scdma, lte-evdo-cdma.**
Ruijie(config-if-Cellular 1)# show cellular info network	Displays the current network access mode.

After the access mode configuration is modified, validate the configuration by using any one of the following three methods: Method 1: Execute **reset** on the interface to reset the interface. Method 2: Execute **shutdown** on the interface, and then execute **no shutdown** after waiting for at least two seconds. Method 3: Save the configuration and restart the host. If the configuration under the 4G interface is modified by remotely logging in to the device through the 4G line, only method 1 or 2 can be used. Any of the three methods can be used if the configuration is not modified through the 4G line.

Example: Select the network access mode.

```
Ruijie(config)#interface cellular 0/0
Ruijie(config-if-Cellular 1)# plmn mode manual lte
Ruijie(config-if-Cellular 1)# show cellular info network
 ------Network Information------
System mode = LTE mode(17)
System submode = HSPA+ mode(9)
Service state = Effective service(2)
```

```
Roam state = Not roaming status(0)
Service domain = EPS service(4)
Cell ID = 134326788
LAC = 32796
```

1.2.4 Associating 4G Interface with Interface Protocol Status

1.2.4.1 Configuration steps

Step	Configuration Task	Description
1	Associated Master Interface Protocol Status (BFD or TRACK)	Optional

1.2.4.2 Configuring Association with Master Interface Protocol Status

Command	Description
Ruijie# configure terminal	Enters the global configuration mode.
Ruijie(config)# interface cellular *0/0*	Enters the interface configuration mode.
Ruijie(config-if-Cellular 1)# apply detect interface vlan 10 {bfd | track track_id}	Configures the protocol status association between the 4G link and primary link: BFD or TRACK.

Example: Configure association with the BFD protocol.

```
Ruijie(config)#interface cellular 0/0
Ruijie(config-if-Cellular 1)# apply detect interface vlan 10 bfd
Ruijie(config-if-Cellular 1)# show running-config interface cellular 0/0
Building configuration...
Current configuration : 65 bytes
!
interface Cellular 0/0
 apply detect interface vlan 10 bfd
```

1.2.5 Associating 4G Interface with Dial Stimulation (Interesting Traffic) Rule

1.2.5.1 Configuration steps

Step	Configuration Task	Description
1	Dial Stimulation (Interesting Traffic) Rule of 4G Interface Association	Optional

1.2.5.2 Configuring Association with Master Interface Protocol Status

Command	Description
Ruijie# configure terminal	Enters the global configuration mode.
Ruijie(config)# interface cellular 0/0	Enters the interface configuration mode.
Ruijie(config-if-Cellular 1)# apply detect dial-list { list_id | list_name [idle-timeout seconds]	Configures the association of the 4G link with the dial stimulation (interesting traffic) rule.

Example: Configure the association with the interesting traffic rule with the ACL ID of 100.

```
Ruijie(config)#interface cellular 0/0
Ruijie(config-if-Cellular 1)# apply detect dial-list 100
Ruijie(config-if-Cellular 1)# show running-config interface cellular 0/0
Building configuration...
Current configuration : 65 bytes
!
interface Cellular 0/0
 apply detect dial-list 100
```

1.2.6 Configuring 4G Backup

1.2.6.1 Configuration steps

Step	Configuration task	Description
1	Configuring 4G Backup Function	Optional

1.2.6.2 Configuring 4G Backup Function

Command	**Description**
Ruijie# configure terminal	Enters the global configuration mode.
Ruijie(config)# interface cellular *0/0*	Enters the interface configuration mode.
Ruijie(config-if-Cellular 1)# apply detect interface vlan 10 {bfd | track track_id}	Configures the protocol status association between the 4G link and primary link: BFD or TRACK.
Ruijie(config-if-Cellular 1)#apply dial-on-demand min-delay delay1 max-delay delay2;	Configures dial-on-demand and configures the random delay access range.

1. The association with master interface protocol status must be configured for dial-on-demand.

2. The 4G backup link will be used only when the protocol status of the primary link is down.

Example: Configure backup-on-demand and set the dial delay range to 1-60s.

```
Ruijie(config)#interface cellular 0/0
Ruijie(config-if-Cellular 1)# apply detect interface vlan 10 bfd
Ruijie(config-if-Cellular 1)# apply dial-on-demand min-delay 1 max-delay 60
uijie(config-if-Cellular 1)# show running-config interface cellular 0/0
Building configuration...
Current configuration : 65 bytes
!
interface Cellular 0/0
 apply detect interface vlan 10 bfd
 apply dial-on-demand min-delay 1 max-delay 60
Ruijie#
```

1.2.7 Configuring Interesting Traffic-Triggered Dialing

1.2.7.1 Configuration steps

Step	Configuration Task	Description
1	Configuring Interesting Traffic-Triggered Dialing	Optional

1.2.7.2 Configuring Dial-on-demand

Command	Description
Ruijie# configure terminal	Enters the global configuration mode.
Ruijie(config)# interface cellular 0/0	Enters the interface configuration mode.
Ruijie(config-if-Cellular 1)# apply detect dial-list { list_id | list_name [idle-timeout seconds]	Configures the association of the 4G link with the dial stimulation (interesting traffic) rule.
Ruijie(config-if-Cellular 1)#apply dial-on-demand	Configures dial-on-demand and configures the random delay access range.

The association with master interface protocol status must be configured for dial-on-demand.

Example: Configure the interesting traffic-triggered dialing function, and use the ACL rule with the ACL ID 100 as the condition for triggering 4G dialing. In other words, the IP packet with any source address and a destination address of 7.7.7.7 will trigger 4G dialing.

```
Ruijie(config)#interface cellular 0/0
Ruijie(config-if-Cellular 1)# apply detect dial-list 100
Ruijie(config-if-Cellular 1)# apply dial-on-demand
Ruijie(config-if-Cellular 1)# show running-config interface cellular 0/0
Building configuration...
Current configuration : 65 bytes
!
ip access-list extended 100
   10 permit ip any host 7.7.7.7
!
interface Cellular 0/0
 apply detect dial-list 100
apply dial-on-demand
Ruijie#
```

1.2.8 Anti-traffic-impact

1.2.8.1 Configuration Steps

Step	Configuration Task	Description
1.	Configure the ACL for limiting traffic.	Mandatory.
2	Configure the anti-traffic-impact function.	Mandatory.

1.2.8.2 Configuring Anti-traffic-impact Function

Command	Description
Ruijie# configure terminal	Enters the global configuration mode.
Ruijie(config)# ip access-list extended 109	Configures the ACL that limits traffic of video servers.
Ruijie(config-ext-nacl)# deny ip any host 192.168.50.246	Limits traffic destined for a specified destination address.
Ruijie(config-ext-nacl)# permit ip any any	Allows other traffic.
Ruijie(config)# interface cellular *0/0*	Enters the interface configuration mode.
Ruijie(config-if-Cellular 1)# apply detect interface vlan 10 {bfd | track track_id}	Configures protocol state association between the 4G link and primary link: BFD or TRACK.
Ruijie(config-if-Cellular 1)# apply traffic-anti-impact interface VLAN 1 list 109	Configures the anti-traffic-impact function.

1. Configure protocol state association of the primary interface to perform dialing on demand.

The anti-traffic-impact function is supported only on 4G mobile routers of the RSR820 series.

Example: Configure dialing on demand and set the dialing delay from 1s to 60s.

```
Ruijie(config)#interface cellular 0/0
Ruijie(config-if-Cellular 1)# apply detect interface vlan 10 bfd
Ruijie(config-if-Cellular 1)# a apply traffic-anti-impact interface VLAN 1 list 109
uijie(config-if-Cellular 1)# show running-config interface cellular 0/0
Building configuration...
Current configuration : 65 bytes
!
interface Cellular 0/0
 apply detect interface vlan 10 bfd
 apply traffic-anti-impact interface VLAN 1 list 109
Ruijie#
```

1.2.9 PIN Code Protection

This function sets a password for a SIM card to prevent unauthorized users from using the SIM card. The PIN code of the SIM card is 1234 by default. Three PIN code protection modes are available, with the differences as follows:

Command	**Use**	**Differences**	**Scenario**
simple	Simple PIN code protection mode: The user needs to use the CLI command to enter the current PIN code of the SIM card. If the PIN code is correct, the simple PIN code protection function is enabled; if the PIN code is wrong, the SIM card cannot be used, and PUK unlocking is needed (for details, refer to operation of the PUK unlocking scenario). Use the NO command to disable the simple PIN code mode.	The numeric string entered by the user is the PIN code corresponding to the SIM card. This PIN code is visible to the PIN code administrator and the security level is low.	The SIM card can be inserted into a router or mobile phone terminal supporting the PIN code input function.
strict-pin	Strict PIN code protection mode: The user needs to use the CLI command to enter the current PIN code of the SIM card and a HASH string. If the PIN code is correct, the strict PIN code protection function is enabled and the HASH string will replace the current PIN code; if the PIN code is wrong, the SIM card cannot be used, and PUK unlocking is needed (for details, refer to operation of the PUK unlocking scenario).	A new PIN code is obtained after the numeric string entered by the user is hashed. The new PIN code is invisible to the PIN code administrator and the security level is high.	The SIM card can only be inserted into Ruijie router for use.
bind-router	Binding the router PIN code protection mode: The user needs to use the CLI command to enter the current PIN code of the SIM card and a HASH string. If the PIN code is correct, the PIN code binding protection function is enabled and the current PIN code is replaced after hashing using the router serial number; if the PIN code is wrong, the SIM card cannot be used, and PUK unlocking is needed (for details, refer to operation of the PUK unlocking scenario).	A new PIN code is obtained after the router serial number is hashed. This PIN code is directly bound with the router and the security level is highest.	The SIM card can only be used on a fixed router of Ruijie.

1.2.9.1 Enabling PIN Code Protection

Command	Description
Ruijie# configure terminal	Enters the global configuration mode.
Ruijie(config)# interface cellular 0/0	Enters the interface configuration mode.
Ruijie(config-if-Cellular 1)# plmn pin-protection simple 0 1234	Enables the simple PIN code protection function. The last numeric string in configuration is the PIN code corresponding to the SIM card.
Ruijie(config-if-Cellular 1)# plmn pin-protection bind-router 0 1234	Enables the PIN code protection function of router binding mode. The last numeric string in configuration is the PIN code corresponding to the SIM card.
Ruijie(config-if-Cellular 1)# plmn pin-protection strct-pin 12345678 0 1234	Enables the strictly encrypted PIN code protection function. The last numeric string in configuration is the PIN code corresponding to the SIM card. The first character string is a strictly encrypted hash string.

1. When a 3G card and the corresponding SIM card are inserted for the first time, the initial password of the SIM card is known by the customer. The password is 1234 by default if it was not changed before.

2. The router administrator enables the PIN code protection function through CLI. When this command is enabled, enter the correct initial PIN code of the SIM card. If the PIN code is wrong. The SIM card will be locked. PUK unlocking is needed. Refer to scenario 1 of PUK unlocking.

3. When the router is bound or the strict encryption mode is enabled, the PIN code has been encrypted and is not the original PIN code 1234 any more, and 1234 cannot be used on other devices (such as a mobile phone) to enable the PIN code protection function. Only the **no plmn pin-protection** command can be used on this router to disable the PIN code protection function.

4. In router binding or strict encryption mode, the new PIN code is encrypted for saving and invisible to the administrator.

5. Each time when the SIM card connects to the network in dial-up connection mode, the router will use the new PIN code to unlock the SIM card. If the PIN code is correct, normal dialing can be implemented and the SIM card can be used to access the network normally.

6. The three modes use the same command to cancel the PIN code protection: **no plmn pin-protection**. This command can be executed only when the PIN code protection command is executed.

7. The SIM card is relocated from router A to router B, router B has been configured with the PIN code protection related command, and a new PIN code is available. There are two cases for SIM card replacement:

Case 1: If the PIN code of SIM card is consistent with the PIN code for the CLI command of router B, the related process is the same as the initial use process. The code can be replaced directly for use. The router first uses the initial password to enable PIN code protection. After the function is enabled, a new PIN code is generated according to the binding or strict encryption algorithm. After that, the router will use the new PIN code to unlock the SIM card each time the 3G card connects to the 3G network in dial-up connection mode. If the PIN code is correct, normal dialing can be implemented and the 3G card can be used to access the network normally.

Case 2: If the PIN code of the SIM card inconsistent with the PIN code for the CLI command of router B, directly replacement may lead to locking of the SIM card. In this case, a designated administrator needs to unlock the SIM card by using the PUK code, and then reset the PIN code (for details, refer to the section about PUK unlocking).

Case 2 above can be handled by using the following three methods:

①: Before replacing the SIM card, use another device (such as a mobile phone) to change the PIN code of SIM card to the initial PIN code configured on the router, and then insert the card into the router for use;

②: Before replacing the SIM card, delete router configuration, insert the new card, and then configure the PIN protection function again;

③: The SIM card is locked after replacement. In this case, the corresponding PUN code of SIM card can be used to unlock it. Refer to the description about the PUK unlocking scenario.

Example: Configure PIN code protection (in router binding mode; the configuration for other modes is similar).

```
Ruijie(config)#interface cellular 0/0
Ruijie(config-if-Cellular 1)# plmn pin-protection bind-router 0 1234
uijie(config-if-Cellular 1)# show running-config interface cellular 0/0
Building configuration...
Current configuration : 65 bytes
!
interface Cellular 0/0
 plmn pin-protection bind-router 0 1234
```

1.2.9.2 Changing PIN Code

Use the PIN code changing command to change the PIN code if required.

Command	Description
Ruijie# configure terminal	Enters the global configuration mode.
Ruijie(config)# interface cellular 0/0	Enters the interface configuration mode.
Ruijie(config-if-Cellular 1)# plmn pin-protection simple 0 1234	Enables the simple PIN code protection function. The last numeric string in configuration is the PIN code corresponding to the SIM card.
Ruijie(config-if-Cellular 1)# plmn modify 2345	Changes the PIN code.

1. The PIN code changing command can only run in the simple mode.

2. To run this command, judge the modem status. Therefore, the PIN code changing command may fail if the modem status is abnormal. A prompt message is displayed in the system log, and the user can try again later. The PIN code may also be changed through a mobile phone.

3. Command execution may fail if the operator’s network is busy or due to other factors.

4. Use this command with caution. When this command is used, it is necessary to save the existing router configuration.

Example: Configure PIN code protection (in router binding mode; the configuration for other modes is similar).

```
Ruijie(config)#interface cellular 0/0
Ruijie(config-if-Cellular 1)# plmn pin-protection bind-router 0 1234
Ruijie(config-if-Cellular 1)# plmn modify 2345
Proceed with modify pin code and write config?[N0] y
pin code modify success !
Ruijie(config-if-Cellular 1)# show running-config interface cellular 0/0
Building configuration...
Current configuration : 65 bytes
!
interface Cellular 0/0
 plmn pin-protection bind-router 0 2345
```

1.2.9.3 PUK Unlocking

The SIM card will be locked when a wrong PIN code is used to enabled PIN code protection. In this case, PUK unlocking is needed, and a new PIN code needs to be set for the SIM card. When the SIM card is in the PUK locking status, the user will be prompted using the system log printing form, and the user can use the sh cell info command to view the SIM card status.

Command	Description
Ruijie# configure terminal	Enters the global configuration mode.
Ruijie(config)# interface cellular 0/0	Enters the interface configuration mode.
Ruijie(config-if-Cellular 1)# plmn puk-unlock 12345678 1234	PUK unlocking; the first character string is the PUK code and the second character string is the new PIN code.

1. The PUK unlocking is required in the following cases:

Case 1: When the PIN code protection function is enabled, the PIN code behind the command is inconsistent with the corresponding PIN code of SIM card. The SIM card will be locked.

Case 2: After the router binding function of PIN code is enabled, the SIM card is removed and placed in another router with the same configuration for use. The SIM card will be locked.

Case 3: After the strict encryption function of PIN code is enabled, the SIM card is removed and placed in another router with different configuration for PIN code protection. The SIM card will be locked.

Case 4: When the SIM card is replaced, the initial password of the SIM card is inconsistent with the initial PIN code configured for the router. The SIM card will be locked.

2. After PUK unlocking, the PIN code protection is disabled in the SIM card.

3. Use the PUK unlocking command with caution. The SIM card will be damaged when the number of wrong PUK code attempts reaches 10. Before unlocking, you are recommended to first ask the operator about the PUK code corresponding to the SIM card. PUK unlocking can also be implemented through a mobile phone or other terminals.

Example: Configure PIN code protection (in router binding mode; the configuration for other modes is similar).

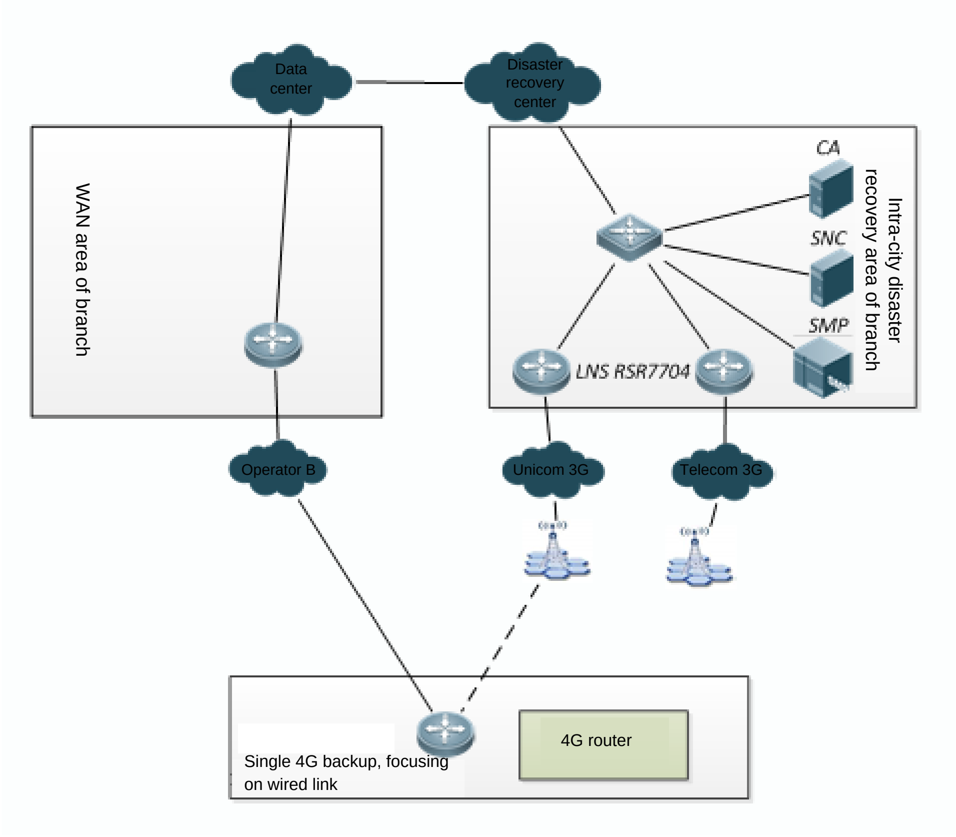
```
Ruijie(config)#interface cellular 0/0
Ruijie(config-if-Cellular 1)# plmn puk-unlock 12345678 1234
```

1.3 Examples of Typical Configuration

1.3.1 Disaster Recovery Scenario of Wired Link and Single 4G Link

1.3.1.1 Network requirements

The scenario involves the wired link and single 4G link disaster recovery. The wired link is used during normal operations. After communication of the wired link is interrupted, the device can automatically detect and then enable the 4G link. The 4G link is disconnected automatically after the wired link is recovered.



Wired and 4G networking topology

1.3.1.2 Configuration steps

Track detection is used in the configuration example. Alternatively, BFD can be used to associate with the wired link. For detailed configuration, refer to the BFD configuration section in the Configuration Manual.

```
(1) Configure the wired link for track detection:
!Here, it is assumed that the IP address of the LNS corresponding to the wired link is 10.10.10.10.
Ruijie(config)#ip rns 1
Ruijie(config-ip-rns-icmp-echo)# icmp-echo 10.10.10.10 out-interface VLAN 1 next-hop 10.10.10.1
Ruijie(config-ip-rns-icmp-echo)# timeout 3000
Ruijie(config-ip-rns-icmp-echo)# frequency 3000
Ruijie(config-ip-rns-icmp-echo)# ntime 3
Ruijie(config)# track 1 rns 1
```

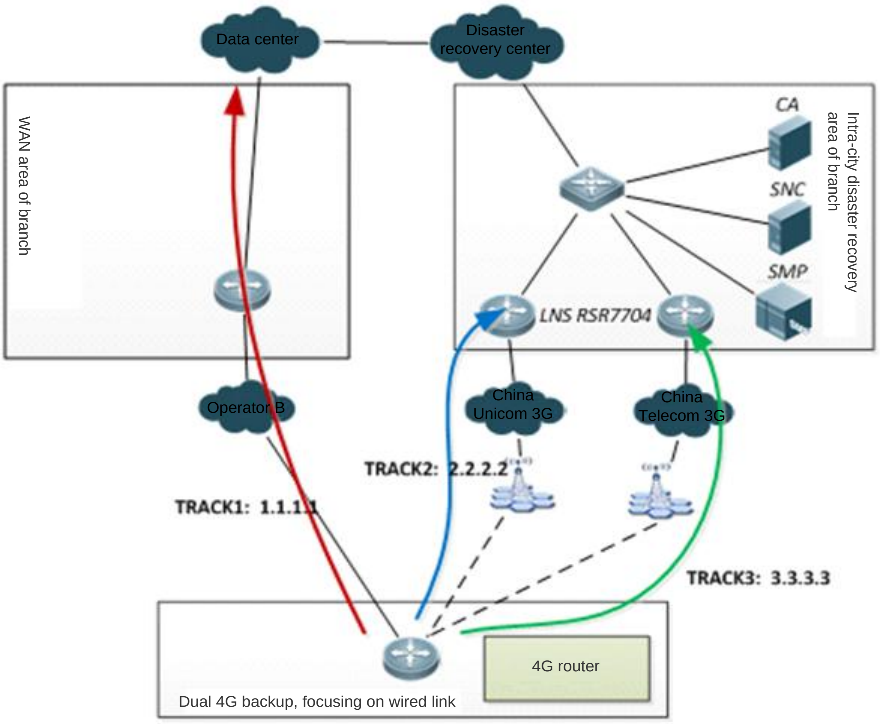
```
(2) Configure 4G dial-on-demand:
Ruijie(config)#interface Cellular 0/0
Ruijie(config-if-Cellular 0/0)# apply detect interface VLAN 1 track 1
Ruijie(config-if-Cellular 0/0)# apply dial-on-demand

(3) Configure the static route:
Ruijie(config)#ip route 0.0.0.0 0.0.0.0 VLAN 1 track 1
Ruijie(config)#ip route 0.0.0.0 0.0.0.0 Cellular 0/0 100
```

1.3.2 Disaster Recovery Scenario of Wired Link and Dual 4G Links

1.3.2.1 Network requirements

The scenario involves the wired link and dual 4G link disaster recovery. The wired link is used during normal operations. After communication of the wired link is interrupted, the device can automatically detect and then enable backup 4G link 1. After communication of backup 4G link 1 is interrupted, the device can automatically detect and then enable backup 4G link 2. Backup 4G link 2 is disconnected automatically after backup 4G link 1 is recovered. Both backup 4G link 1 and backup 4G link 2 are disconnected after the wired link is recovered.

Note: The RSR820 series mobile routers do not support the use method of this scenario.



Wired link and dual 4G links networking topology

1.3.2.2 Configuration steps

Track detection is used in the configuration example. Alternatively, BFD can be used to associate with the wired link. For detailed configuration, refer to the BFD configuration section in the Configuration Manual.

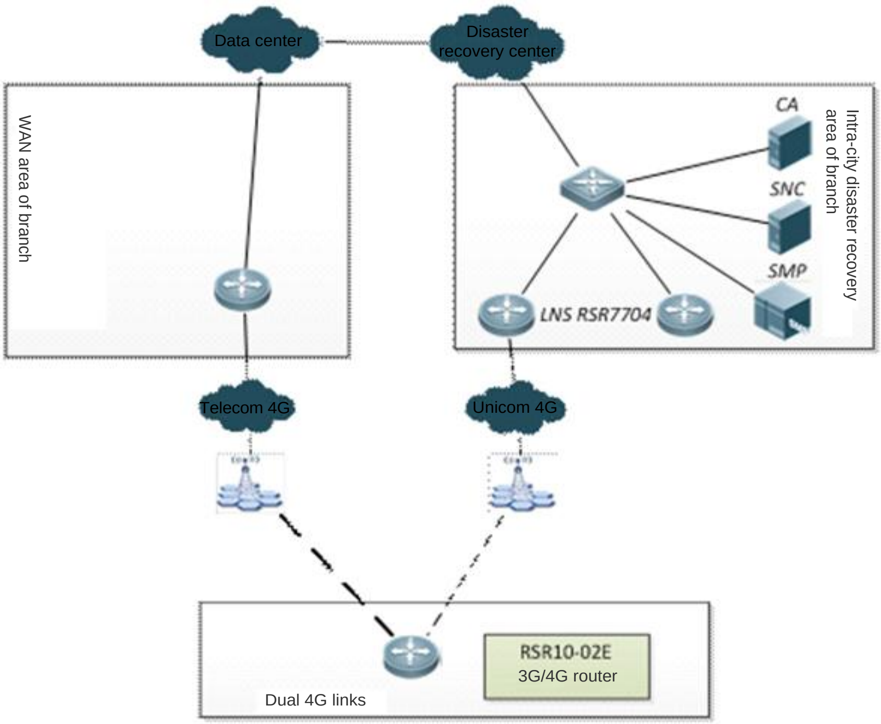
```
(1) Configure the wired link for track detection:
!Here, it is assumed that the IP address of the LNS corresponding to the wired link is 1.1.1.1.
Ruijie(config)#ip rns 1
Ruijie(config-ip-rns-icmp-echo)# icmp-echo 1.1.1.1 out-interface VLAN 1 next-hop 1.1.1.1
Ruijie(config-ip-rns-icmp-echo)# timeout 3000
Ruijie(config-ip-rns-icmp-echo)# frequency 3000
Ruijie(config-ip-rns-icmp-echo)# ntime 3
Ruijie(config)# track 1 rns 1

!Here, it is assumed that the IP address of the LNS corresponding to backup 4G link 1 is 2.2.2.2.
Ruijie(config)#ip rns 2
Ruijie(config-ip-rns-icmp-echo)# icmp-echo 2.2.2.2 out-interface Cellular 0/0
Ruijie(config-ip-rns-icmp-echo)# timeout 3000
Ruijie(config-ip-rns-icmp-echo)# frequency 3000
Ruijie(config-ip-rns-icmp-echo)# ntime 3
Ruijie(config)# track 2 rns 2


(2) Configure 4G dial-on-demand:
Ruijie(config)#interface Cellular 0/0
Ruijie(config-if-Cellular 0/0)# apply detect interface VLAN 1 track 1
Ruijie(config-if-Cellular 0/0)# apply dial-on-demand

Ruijie(config)#interface Cellular 1/0
Ruijie(config-if-Cellular 1/0)# apply detect interface VLAN 1 track 1
Ruijie(config-if-Cellular 1/0)# apply detect interface Cellular 0/0 track 2
Ruijie(config-if-Cellular 1/0)# apply dial-on-demand min-delay 60 max-delay 120

(3) Configure the static route:
ip route 0.0.0.0 0.0.0.0 VLAN 1 track 1
ip route 0.0.0.0 0.0.0.0 Cellular 0/0 100 track 2
ip route 0.0.0.0 0.0.0.0 Cellular 1/0 200 track 3
```

1.3.3 Scenario of Interesting Traffic-Triggered Dialing through Dual 4G Links

1.3.3.1 Network requirements

In this scenario, dialing is triggered by the interesting traffic over the dual 4G links. The active 4G link is used during normal operations, and the active 4G link is in the non-dialing status when there is no interesting traffic. When there is interesting traffic, dialing of the active 4G link is triggered. After communication of the active 4G link is interrupted, the device can automatically detect and then enable the backup 4G link. The backup 4G link needs interesting traffic to trigger dialing. The active 4G link is used preferably when the active 4G link is recovered from the fault.

Note: The RSR820 series mobile routers do not support the use method of this scenario.



Networking topology of interesting traffic-triggered dialing over dual 4G links

1.3.3.2 Configuration steps

```
(1) Configure the active 4G link for track detection:
!Here, it is assumed that the IP address of the LNS corresponding to backup 4G link 1 is 2.2.2.2.
Ruijie(config)#ip rns 2
Ruijie(config-ip-rns-icmp-echo)# icmp-echo 2.2.2.2 out-interface Cellular 0/0
Ruijie(config-ip-rns-icmp-echo)# timeout 3000
Ruijie(config-ip-rns-icmp-echo)# frequency 3000
Ruijie(config-ip-rns-icmp-echo)# ntime 3
Ruijie(config)# track 2 rns 2

(2) Configure the rule of interesting traffic-triggered dialing:
Ruijie(config)#ip access-list extended 100
Ruijie(config-ext-nacl)# 10 permit ip 124.124.124.0 0.0.0.255 any
Ruijie(config-ext-nacl)# 20 permit ip 126.126.126.0 0.0.0.255 any
Ruijie(config-ext-nacl)#!
Ruijie(config-ext-nacl)#!
Ruijie(config-ext-nacl)#ip access-list extended 199
Ruijie(config-ext-nacl)# 10 deny icmp any any
Ruijie(config-ext-nacl)# 20 permit ip any any
Ruijie(config-ext-nacl)#!

(3) Configure the 4G interesting traffic-triggered dialing:
Ruijie(config)#interface Cellular 0/0
Ruijie(config-if-Cellular 0/0)# apply detect dial-list 100
Ruijie(config-if-Cellular 0/0)# apply dial-on-demand

Ruijie(config)#interface Cellular 1/0
Ruijie(config-if-Cellular 0/0)# apply detect dial-list 199
Ruijie(config-if-Cellular 1/0)# apply detect interface Cellular 0/0 track 2
Ruijie(config-if-Cellular 1/0)# apply dial-on-demand min-delay 60 max-delay 120

(3) Configure the static route:
Ruijie(config)#ip route 0.0.0.0 0.0.0.0 Cellular 0/0 track 2
Ruijie(config)#ip route 0.0.0.0 0.0.0.0 Cellular 1/0 100
```

2 Configuring Dialup

2.1 PPPoE Dialup

2.1.1 PPPoE Overview

Ruijie Device supports PPP running on the Ethernet (PPPoE, PPP over Ethernet) interface for DDR. Its characteristics are like the DDR: stimulating dialup if there is data communication and automatically disconnecting the line when it is idle for the specified period.

The PPPoE implementation on Ruijie devices is similar to DDR Profiles, which binds the Ethernet interface to the logical interface and performs the negotiations on the logical interface.

2.1.2 PPPoE Configuration Tasks

From the implementation of PPPoE, the PPPoE configuration tasks include:

- Configuring the Ethernet interface
- Configuring the logical interface
- Configuring the necessary global parameters

2.1.2.1 Configuring the Ethernet interface

The configuration of the Ethernet interface includes:

- Enable the PPPoE on the interface
- Bind the Ethernet interface to the specified dialer pool

Some basic configurations of the Ethernet interface are also required, such as the interface activation (**no shutdown**).

Use the following command to enable PPPoE on the Ethernet interface in interface configuration mode.

Command	Function
Ruijie(config-if)# **pppoe enable**	Enables PPPoE.

Since PPPoE is implemented via DDR Profiles, it is required to bind the Ethernet interface to the specified dialer pool available for the use of the logical interface in DDR Profiles. Use the following commands to bind the Ethernet interface to the specified dialer pool in interface configuration mode.

Command	Function
Ruijie(config-if)# **pppoe-client dial-pool-number** *pool-number* **dial-on-demand**	Binds the Ethernet interface to the specified logical dialer pool (to enable the DDR function. The dialup to the PPPoE server is activated only when there is some message).
Ruijie(config-if)# **pppoe-client dial-pool-number** *pool-number* **no-ddr**	Binds the Ethernet interface to the specified logical dialer pool (to enable automatic dialup to the PPPoE server).

📖 One of **dial-on-demand** and **no-ddr** must be selected. **dial-on-demand** enables the PPPoE DDR function to disconnect the line when the specified idle period expires; **no-ddr** enables the PPPoE auto-dialup function to dial to the PPPoE server automatically.

2.1.2.2 Configuring the logical interface

All PPP negotiations of the PPPoE dialup are based on the logical interface, so it is required to configure the PPPoE dialup related parameters on the logical interfaces.

Use the following commands to configure the logical interface in global configuration mode.

Command	**Function**
Ruijie(config)# **interface dialer** *dialer-number*	Enters the specified logical interface.
Ruijie(config-if)# **ip address negotiate**	Obtains address from negotiation .
Ruijie(config-if)# **dialer pool** *pool-number*	Associates the dialer pool, with the one in the Ethernet interface one-by-one.
Ruijie(config-if)# **dialer idle-timeout** *seconds*	(Optional) Configures the timeout time to disconnect the line when the specified idle time times out.
Ruijie(config-if)# **encapsulation ppp**	Encapsulates PPP because PPPoE is based on PPP.
Ruijie(config-if)# **mtu 1488**	Configures the maximum transmission unit as 1488.
Ruijie(config-if)# **dialer-group** *dialer-group-number*	Sets the dialer group associated with the dialup activation rule.
Ruijie(config-if)# **ppp pap sent-username** *username* **password** *password*	Configures the username and password for authentication.

📖 Except for the command in step 4, all other commands are required. Especially in step 6, the MTU value must be 1488 for normal communication.

If the IP address of the logical interface is modified after a PPPoE connection has been set up, run the **clear pppoe tunnel** command to trigger PPPoE re-negotiation for the modification to take effect in privileged mode.

2.1.2.3 Configuring the necessary global parameters

The necessary PPPoE global parameters include:

- Define the dialup activation rule
- Configure the dialup route

To use PPPoE with other functions, such as NAT, it is required to configure other global parameters. The global parameters must be configured according to the actual conditions.

Use the following command to define the dialup activation rule in global configuration mode.

Command	**Function**
Ruijie(config)# **dialer-list** *dialer-group- number* **protocol** *protocol-name* {**permit** | **deny** | **list** *access-list-number* }	Defines the dialup activation rule.

Since the IP address of the interface is generally obtained from the negotiation and there is no directly-connected route of the logical interface, it is required to configure a dialup route to the destination for the logical interface with the PPPoE dialup, so that the data can be forwarded via the PPPoE interface.

Use the following command to configure the dialup route in global configuration mode.

Command	Function
Ruijie(config)# **ip route 0.0.0.0 0.0.0.0 dialer** *dialer-number* [**permanent**]	Sets the default route. The **permanent** option enables the route always valid, even when the logical interface is in the line invalid period (enable-timeout) (here, the logical interface is in the down status).

📖 It is also possible that the dialup route is not configured as the default route, and the specified route is configured as required. In whatever conditions, a dialup route must be configured to make dialup possible.

2.1.3 Monitoring PPPoE

Use the following commands to monitor the PPPoE in privileged EXEC mode.

Command	Function
Ruijie# **show pppoe** {**tunnel** | **session**}	Shows the PPPoE configuration.
Ruijie# **show interfaces dialer** *dialer-number*	Shows the logical interface configuration.
Ruijie# **debug pppoe** {**datas**|**errors**|**events**|**packets** }	Turns on the PPPoE debug switch.

2.2 Callback

2.2.1 Callback Overview

2.2.1.1 Introduction

The callback needs the client/server relation between the two parties of the dialup. One remote callback client dials to connect the callback server, and the callback server performs authentication for the dial-in user. If the authentication passes, the callback server uses the related information of the remote host for callback.

📖 If the callback fails due to busy line, no reply, or other reasons, no redialing is done. If the callback server has no available interface for callback, no retry will be done.

Callback client: remote device or host that requests the callback

Callback server: the device that accepts the callback request, and if the authentication passes, disconnects the current connection and locally initiates the dialing to the remote client.

2.2.1.2 Purposes

The callback is generally used in the following three scenarios:

- Save cost: The call charge may be different for different areas. For example, the dialing from village to city may be more expensive than the reverse direction, so it is possible to save cost by calling back from the city to village.

- Unified call charge: An organization may have quite a lot of branches. It is possible to call back from the center to the branches so that the calls are all paid by the center to facilitate the financial management and statistics.
- Security considerations: The callback numbers set for the callback server are the reliable and valid numbers that have been configured. This helps ensure the location of the callback client is valid, forbidding the invalid dialing from the range out of control.

2.2.1.3 Callback configuration tasks

Before configuring the callback, make sure the global DDR configuration preparations are completed. The callback configuration tasks involve:

- Configure the callback client
- Configure the callback server

Configuring the callback client

Use the following commands to configure the callback client for the device in global configuration mode.

Command	**Function**
Ruijie(config)# **interface** *type number*	Specifies the interface and enters the interface configuration mode.
Ruijie(config-if)# **dialer in-band**	Enables DDR.
Ruijie(config-if)# **encapsulation ppp**	Encapsulates PPP.
Ruijie(config-if)# **ppp authentication** {**chap** | **pap**}	Configures the PPP authentication.
Ruijie(config-if)# **dialer map** *protocol next-hop-address* **name** *hostname dial-string*	Maps the callback server hostname, address and telephone number.
Ruijie(config-if)# **ppp callback request**	Configures the current interface as the callback client.
Ruijie(config-if)# **dialer hold-queue** *[packets* [**timeout** *seconds*]]	(Optional) Configures the hold queue on the interface.
Ruijie(config-if)# **dialer-group** *group*	Associates the dialup activation rule.
Ruijie(config-if)# **async mode dedicated**	Enables automatic negotiation of the dialup mode.

Configuring the callback server

Use the following commands to configure the callback server for the device in global configuration mode.

Command	**Function**
Ruijie(config)# **interface** *type number*	Specifies the interface and enters the interface configuration mode.
Ruijie(config-if)# **dialer in-band**	Enables DDR.
Ruijie(config-if)# **encapsulation ppp**	Encapsulates PPP.
Ruijie(config-if)# **ppp authentication** {**chap** | **pap**}	Configures the PPP authentication.
Ruijie(config-if)# **dialer map** *protocol next-hop-address* **name** *hostname* **class** *classname dial-string*	Maps the callback client hostname, address and telephone number.
Ruijie(config-if)# **dialer hold-queue** *[packets]* [**timeout**	(Optional) Defines the hold queue on the interface.

Command	Function
seconds]]	
Ruijie(config-if)# **dialer enable-timeout** *seconds*	(Optional) Configures the line invalid waiting time (i.e. waiting time before the callback).
Ruijie(config-if)# **ppp callback accept**	Accepts the callback request.
Ruijie(config-if)# **dialer-group** *group*	Associates the dialup activation rule.
Ruijie(config-if)# **async mode dedicated**	Enables automatic negotiation of the dialup mode.
Router(config-pmap)# **exit**	Returns to the global configuration mode.
Ruijie(config)# **map-class dialer** *classname*	Defines the dialup mapping class and enters the mapping class configuration mode.
Ruijie(config-map-class)# **dialer callback-server** [**username**] [**dial-string**]	Configures the dialup mapping class as the callback server.

🕮 On the callback server, **dialer enable-timeout** controls the callback start time. If not configured, the default time is used, namely 15 seconds.

2.2.2 Monitoring Callback

Use the following command to monitor the callback in privileged EXEC mode.

Command	Function
Ruijie # **debug dialer packet**	Turns on the DDR debug switch.

2.3 Dialer Watch

2.3.1 Dialer Watch Overview

2.3.1.1 Introduction

The principle of the Dialer Watch is that the local route in the device is detected, and the backup dialup interface is triggered to dial if the specified route monitored does not exist in the routing table. If the specified route monitored appears in the routing table, the backup dialup interface will be triggered to disconnect the dialup line.

2.3.1.2 Purposes

For the general dialup backup, when the master line is disconnected, the backup link just turns from the standby status to the spoof up status but this does not indeed trigger the dialup. The dialup is triggered only when there is some traffic stimulation.

The purpose of the Dialer Watch is that it triggers the dialup as long as the monitored route disappears, freeing away the dependence of the traffic stimulation.

2.3.2 Dialer Watch Configuration Tasks

Before configuring the dialer watch, make sure the global DDR configuration preparations are completed. The dialer watch configuration tasks involve:

- Configure the backup interface for the DDR dialup
- Configure the monitored route for the dialup
- Configure how long to trigger dialup after the monitored route disappears
- Configure how long to disconnect the dialup after the monitored route appears

2.3.2.1 Configuring dialer watch

Use the following commands to configure the dialer watch in global configuration mode.

Command	**Function**
Ruijie(config)# **interface** *type number*	Specifies the interface and enters the interface configuration mode.
Ruijie(config-if)# **dialer watch-group** *group-number*	Enables dialer watch on the backup line.
Ruijie(config)# **dialer watch-list** group-number **ip** *ip-address address-mask*	Defines all the routes to be monitored. The dialer watch function can monitor multiple routes.
Ruijie(config)# **dialer watch-list** group-number **delay** {**connect** *connect-time* }	Configures how long to trigger dialup on the backup interface after the monitored route disappears. If not configured, the dialup is triggered immediately by default.
Ruijie(config)# **dialer watch-list** group-number **delay** {**disconnect** *disconnect-time* }	Configures how long to disconnect the dialup on the backup interface after the monitored route appears. By default, it is disconnected immediately.

Reliability Configuration

1. Configuring RLDP
2. Configuring MSTP
3. Configuring LLDP
4. Configuring VRRP
5. Configuring Hot Swapping
6. Configuring Multilink Gateway Load Balancing
7. Configuring Rns & track

Configuring RLDP

Overview

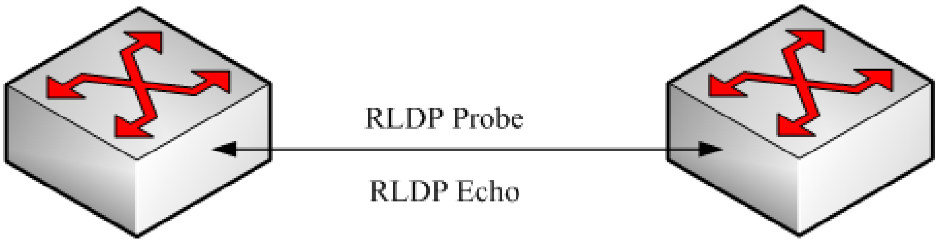
The Rapid Link Detection Protocol (RLDP) is one of Ruijie's proprietary link protocol designed to detect Ethernet link fault quickly.

General Ethernet link detection mechanism only makes use of the status of the physical connections and detects the connectivity of the link via the auto-negotiation of the physical layer. This detection mechanism has restrictions and sometimes cannot provide reliable link detection information for the user. For example, if the optical fiber receiving line pair on the optical interface is misconnected, due to the existence of the optical converter, the related port of the device is "linkup" physically but actually the corresponding layer-2 link cannot work for communications. Here is another example. There is an intermediate network between two Ethernet devices. Due to the existence of the network transmission relay devices, the same problem may occur if those relay devices are faulty.

The RLDP enables easy detection of Ethernet device link fault, including the one-way link fault, two-way link fault and loop link fault.

The RLDP implements the detection by exchanging the RLDP messages at the two ends of the link, as shown in Figure-1:

Figure-1:



The RLDP defines two protocol messages: Probe message and Echo message. The RLDP sends the Probe message of this port to the port with RLDP configured and in linkup status on regular basis, and waits for the Echo message from the neighbor port and waits for the Probe message sent by the neighbor ports. If a link is correct both physically and logically, a port shall be able to receive the Echo message of the neighbor port as well as the Probe message of the neighbor port. Otherwise, the link is considered abnormal.

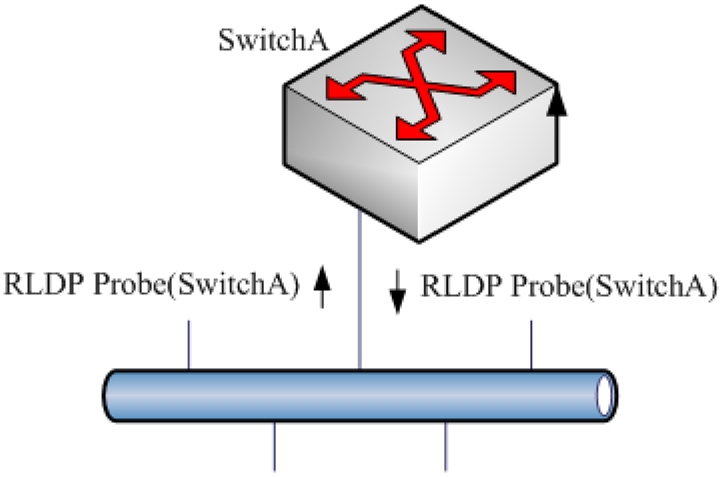
Note

To make use of the one-way detection and two-way detection functions of the RLDP, it is necessary to ensure the RLDP is enabled on the ports at both ends of the link. And, it is not allowed for a port with RLDP enabled to connect multiple neighbor ports. Otherwise, the RLDP cannot detect the health conditions of every neighbor link.

Typical Application

Loop detection:

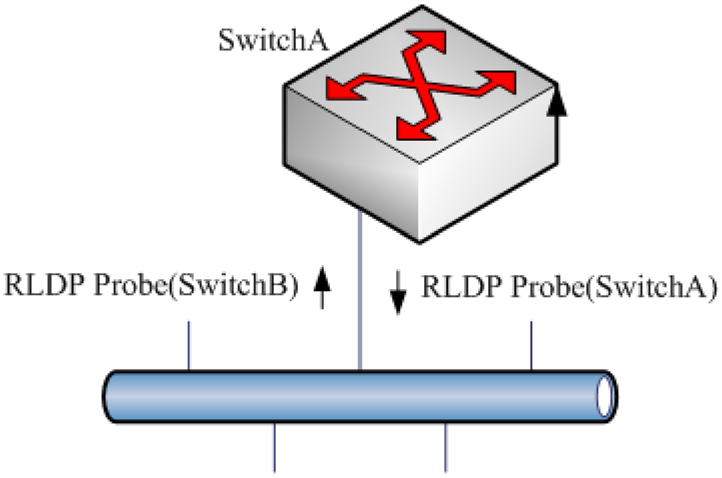
Figure-2: Loop detection



The so-called loop fault means that a loop appears on the links connected with the port. A shown above, on a port the RLDP receives the RLDP message sent from its machine, so the port is considered as loop fault. So, the RLDP deals with the fault according to the user configurations, including alarming, setting port violation, turning off the SVI with that port, turning off the port learning forwarding, and more.

One-way link detection:

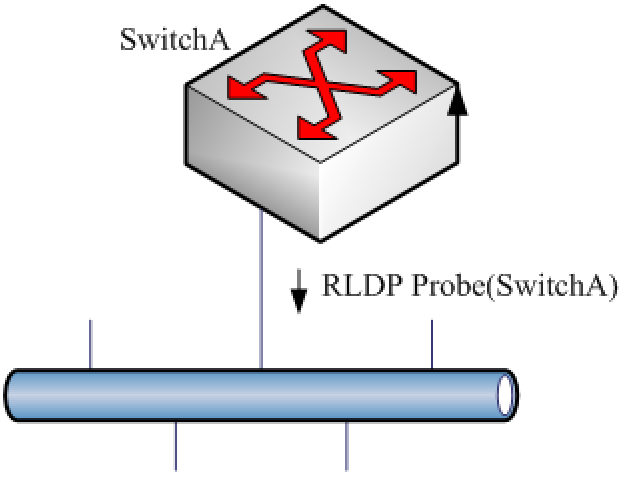
Figure-3: One-way link detection



The so-called one-way link detection means the link connected with the port can receive message only or send messages only (due to misconnection of the optical receiving line pair, for example). As shown above, the RLDP only receives the detection message from the neighbor port on a port, so it is considered one-way link fault. So, the RLDP deals with the fault accordingly according to the user configurations. In addition, if the port cannot receive any RLDP detection message, it is also considered one-way link fault.

Two-way link detection:

Figure-4:Two-way link detection



This means that fault occurs at the frame transmission/receiving at both ends of the link. As shown above, the port of the device sends the RLDP probe message but has never received the Echo message or the Probe message from the neighbors. So, it is considered two-way link fault. From the nature of the fault, the two-way fault actually includes the one-way fault.

Note

If the party at one of the two link ends has not enabled the RLDP, the diagnosis also shows two-way or one-way link fault. So, in configuring two-way link detection or one-way link detection, the administrator shall make sure that the RLDP is enabled at both ends to avoid the incorrect diagnosis information.

Default Configuration

Global RLDP status	Disabled
Port RLDP status	Disabled
Detection interval	3 seconds
Maximum detection times	2

Caution

The RLDP can be configured only on the basis of the switching interface (including AP) and the routing interface.

Caution All RLDP frames are untagged.

Caution In the RLDP fault processing type, the block function and the STP are mutually exclusive. In other words, if the fault processing type configured on the port is "block", it is recommended to disable STP; otherwise, since the STP cannot recognize one-way link, possibly the STP allows port forwarding but the RLDP is configured with port blocking.

Configuring RLDP

Enabling RLDP Globally

The RLDP works on the port only when the global RLDP is enabled.

In global configuration mode, follow these steps to enable RLDP:

Command	Function
Ruijie(config)# **rldp enable**	Enables global RLDP function.
Ruijie(config)# **end**	Returns to privileged EXEC mode.

Use the **no** form of this command to disable global RLDP.

Configuring RLDP on the Port

The RLDP operation is port-based, so the user needs to explicitly configure which ports shall run RLDP. In configuring the port RLDP, it is required to specify the diagnosis type and the troubleshooting method for the port at the same time. The diagnosis types include unidirection-detect, bidirection-detect and loop-detect. The troubleshooting methods include warning, block, shutdown-port, and shutdown-svi.

In global configuration mode, follow these steps to configure the RLDP on the port:

Command	Function
Ruijie(config)# **interface** *interface-id*	Enters interface configuration mode.
Ruijie(config-if)# **rldp port** {**unidirection-detect** | **bidirection-detect** | **loop-detect** } {**warning** | **shutdown-svi** | **shutdown-port** | **block**}	Enables RLDP on the port and configures the diagnosis type and troubleshooting method at the same time.
Ruijie(config-if)# **end**	Returns to privileged EXEC mode.

The **no** form of this command disables RLDP on the port and the configured detection types one by one.

In the example below, the RLDP is configured on GigabitEthernet 0/5, and multiple diagnosis types and troubleshooting methods are specified:

```
Ruijie# configure terminal
Ruijie(config)# interface gigabitEthernet 0/5
Ruijie(config-if)# rldp port unidirection-detect
shutdown-svi
Ruijie(config-if)# rldp port bidirection-detect warning
Ruijie(config-if)# rldp port loop-detect block
Ruijie(config-if)# end
Ruijie# show rldp interface gigabitEthernet 0/5
port state     : normal
local bridge    : 00d0.f822.33ac
neighbor bridge : 0000.0000.0000
neighbor port   :
unidirection detect information:
action : shutdown svi
state  :  normal
bidirection detect information :
action : warnning
state  :  normal
loop detect  information    :
action : block
state  :  normal
```

Several precautions in configuring port detection:

- The routing interface does not support the shutdown-svi error handling method, so this method is not executed in case of the occurring of detection error.
- In configuring loop detection, the neighbor devices downward connected with the port cannot enable the RLDP detection; otherwise, the port cannot have correct detection.
- If the block method is configured on the aggregated port and the link detection error happens, do not change the member port relations of the aggregate port before the port reset detection; otherwise, the forwarding status of the member interface may have unexpected effects of forwarding status.
- If the RLDP detects link error, alarm information will be given. The user can send the alarm information to the log server by configuring the log function. At least 3 levels of log shall be ensured.
- You are recommended to specify the diagnosis type of the loop detection to shutdown-port for the reason that for some devices, even if the device detects the loop and specifies the block port, a large amount of packets will be sent to the CPU for the hardware chip limitation.
- If you configure RLDP on AP port, you are recommended to specify the diagnosis type of loop detection to shutdown-port.

Configuring RLDP Detection Interval

The port with the RLDP function enabled will send the RLDP Probe messages on a regular basis.

In global configuration mode, follow these steps to configure the RERP detection interval:

Command	**Function**
Ruijie(config)# **rldp detect-interval** interval	Configures the detection interval within the range 2-15s, 3s by default.
Ruijie(config)# **end**	Returns to privileged EXEC mode.

The **no** form of the command restores the value to its default.

Configuring the Maximum RLDP Detection Times

If the port with RLDP enabled cannot receive messages from neighbors in the maximum detection period (maximum detection times X detection interval), that port will be diagnosed as faulty. See the Overview for details of the fault types.

In global configuration mode, follow these steps to configure the RERP maximum detection times:

Command	**Function**
Ruijie(config)# **rldp detect-max** Num	Configures the maximum detection times, num range 2-10, 2 by default.
Ruijie(config)# **end**	Returns to privileged EXEC mode.

The **no** form of this command restores the value to its default.

Note

The maximum detection times only take effect in the unidirectional link detection and bidirectional link detection, and will not take effect if only loop detection is enabled on a port.

Restoring the RLDP Status of the Port

The port with shutdown-port troubleshooting method configured cannot resume the RLDP detection actively after a fault occurs. If the user confirms the fault removed, run the recovery command to restart the RLDP on the shutdown port. This command sometimes may make the other ports with detection errors resume.

In privileged EXEC mode, follow these steps to resume the RLDP detection of the port:

Command	**Function**
Ruijie# **rldp reset**	Makes any port with RLDP detection failure resume the detection.

Note

The **errdisable recover** command can be used in the global configuration mode to restart, instantly or at fixed time, the RLDP detection of the port that is set violation by RLDP. It is worth mentioning that when there are some relay devices between rldp ports, if you use **errdisable recover interval** to restore the fault timely, you need to set the value of rldp detection time greater than that of **errdisable recover interval**, that is, the value of detect-interval* detect-max total time is greater than that of **errdisable recover interval** to prevent error judgment.

Enabling RLDP Neighbor Negotiation

RLDP neighbor negotiation is disabled by default.

In global configuration mode, follow these steps to configure RLDP neighbor negotiation:

Command	Function
Ruijie(config)# **rldp neighbor-negotiation**	Enables RLDP neighbor negotiation.
Ruijie(config)# **no rldp neighbor-negotiation** Ruijie(config)# **default rldp neighbor-negotiation**	Disables RLDP neighbor negotiation.

Note

With neighbor negotiation enabled, RLDP unidirectional-/bidirectional-link detection starts only after the neighbor negotiation is successful. (Receiving the Prob message from the neighbor indicates the neighbor negotiation is successful.)

Displaying RLDP Configuration

Displaying the RLDP Status of All Ports

In privileged EXEC mode, run the following commands to display the RLDP global configuration and the port detection information with RLDP detection configured:

Command	Function
Ruijie# **show rldp**	Displays the RLDP global configuration and the port detection information with RLDP detection configured

In the example below, the **show rldp** command is used to display the detection information of all RLDP ports:

```
Ruijie# show rldp
rldp state          : enable
rldp hello interval     : 2
rldp max hello      : 3
rldp local bridge   : 00d0.f8a6.0134
```

```
------------------------------------------------
interface GigabitEthernet 0/1
port state:normal
neighbor bridge     : 00d0.f800.41b0
neighbor port   : GigabitEthernet 0/2
unidirection detect information:
action  : shutdown svi
state   : normal

interface GigabitEthernet 0/24
port state:error
neighbor bridge     : 0000.0000.0000
neighbor port   :
bidirection detect information :
action  : warnning
state   : error
```

As shown above, port GigabitEthernet 0/1 is configured with unidirection detection. No error is detected now, and the port status is normal. Port GigabitEthernet 0/24 is configured with bidirection detection, and bidirection fault is detected.

Displaying the RLDP Status of the Specified Port

In privileged EXEC mode, run the following command to display the RLDP detection information of the specified port:

Command	Function
Ruijie# **show rldp interface** *interface-id*	Displays the RLDP detection information of interface-id.

In the example below, the **show rldp interface GigabitEthernet** *0/1* command is used to display the RLDP detection information of port fas0/1:

```
Ruijie# show rldp int GigabitEthernet 0/1
port state     :error
local bridge   : 00d0.f8a6.0134
neighbor bridge : 00d0.f822.57b0
neighbor port   : GigabitEthernet 0/1
unidirection detect information:
action: shutdown svi
state :  normal
bidirection detect information :
action : warnning
state :  normal
loop detect  information    :
action: shutdown svi
state :  error
```

As shown above, the port GigabitEthernet 0/1 is configured with three detection types: unidirection detection, bidirection detection and loop detection. The troubleshooting methods are shutdown-svi and warning. Error is found in loop detection so the current port status is error. Accordingly, the SVI of the port is shutdown.

Configuring MSTP

STP and RSTP Overview

Ruijie series supports both the STP protocol and the RSTP protocol, as well as complying with the IEEE 802.1D and IEEE 802.1w standards.

The STP protocol can prevent broadcast storm caused by link loops and provide link redundancy and backup.

For the layer 2 Ethernet, there is only one active channel between two LANs to avoid broadcast storm. However, it is necessary to set up redundant links to improve the reliability of a LAN. Furthermore, some channels should be in the backup status in order to take up its work when a link fails. It is obviously hard to control this process by manual. The STP protocol can complete this work automatically. It enables a device in a LAN to:

- Discover and activate an optimal tree-type topology of the LAN.
- Detect and fix failures and automatically update the network topology to offer the possible optimal tree-type structure at any time.

The LAN topology is automatically calculated by a set of bridge parameters set by the administrator. The proper configuration of these parameters is helpful to offer an optimal solution.

The RSTP protocol is completely compatible with the 802.1D STP protocol downward. As with traditional protocol, the RSTP protocol can prevent loop and offer link redundency. The most critical feature of the RSTP protocol is quickness. If the bridges in a LAN support the RSTP protocol and are configured appropriately by administrators, it will take no more than 1 second to re-span the topology tree once the network topology changes (it takes about 50 seconds for traditional STP protocol to re-span the topology tree).

Caution

For the switch buffer control, see the chapter *Buffer Control* in *Configuring QOS*.

Bridge Protocol Data Units (BPDU):

A stable tree-type topology depends on the following elements :

- The unique bridge ID of each bridge consists of the bridge priority and the MAC address.
- The root path cost refers to the cost from a bridge to the root bridge.
- Each port ID consists of the port priority and port number.

By exchanging the Bridge Protocol Data Units (BPDU) frame destined to the multicast address 01-80-C2-00-00-00 (in hex), bridges gets the information necessary for building the optimal tree-type topology.

A BPDU is comprised of the following elements:

- Root Bridge ID (root bridge ID that a bridge considers)
- Root Path cost (Root Path cost of a bridge).
- Bridge ID (ID of a bridge).

- Message age (the live time of the message)
- Port ID (port ID sending the message).
- Forward-Delay Time, Hello Time and Max-Age: time parameters.
- Other flag bits, such as network topology change and port status.

Once a port of a bridge receives a BPDU message whose priority is higher than its priority (or smaller bridge ID and smaller root path cost), the bridge will store this message on the port while updating and propagating them to all other ports. If the BPDU with lower priority is received, the bridge will discard this message.

This mechanism propagates a BPDU message of higher priority in the whole network. As a result:

- A bridge is elected to be the root bridge in the network.
- Each bridge other than the root bridge has a root port that offers a shortest path to the root bridge.
- Each bridge will calculate the shortest path to the root bridge.
- Each LAN has a designated bridge that lies in the shortest path between this LAN and the root bridge. The port for connecting the designated bridge and the LAN is referred to as the designated port.
- The root port and the designated port are in the forwarding status.
- Other ports beyond the spanning tree are in the discarding status.

Bridge ID

As specified in IEEE 802.1W standard, each bridge has an unique bridge ID based on which the root bridge is elected in spanning tree algorithm. The bridge ID consists of eight bytes, in which the last six bytes are the MAC address of the bridge, and the first two bytes are shown in the table below. Of which, the first four bits denote the priority, while the last twelve bits denote the system ID for extending the protocol in the future. This value is 0 in the RSTP, so the priority of the bridge should be configured as the multiple of 4096.

	Priority value				**System ID**											
Bit	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
Value	32768	16384	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1

Spanning-Tree Timers

The following describes three timers impacting the performance of spanning tree.

- Hello timer: Interval to send the BUDU message.
- Forward-Delay timer: Interval to change the port status, that is, the time interval at which the port switches from the listening status to the learning status and vice versa when the RSTP protocol runs in the compatible STP protocol mode.
- Max-Age timer: The longest time for the BPDU message. The system will discard the message when the timer times out.

Port Roles and Status

A port plays a role to present its function in the network topology.

- Root port: The port that provides the shortest path to the root bridge.
- Designated port: The port through which each LAN is connected to the root bridge.

- Alternate port: The alternate port of the root port that will take up its work when the root port fails.
- Backup port: The backup port of the designated port. If two ports of a bridge are connected to a LAN, the port with higher priority is the designated port and the other one is the backup port.
- Disable port: The port that is not in the active status, namely, the ports whose operation status is down.

Figure 1, Figure 2 and Figure 3 below show the roles of various ports:

R = Root port D = Designated port A = Alternate port B = Backup port

Unless otherwise stated, the priorities of these ports are in the descending order from left to right.

Figure-1

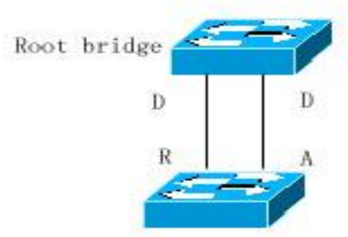


Figure-2

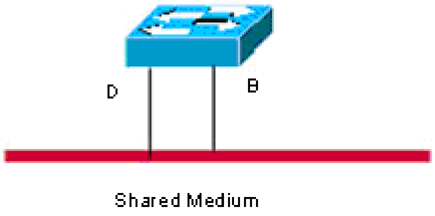
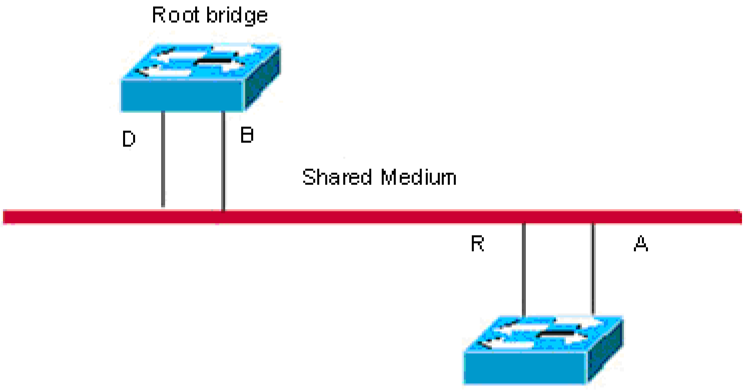


Figure-3



There are three port states for every port to indicate whether the data packet is forwarded and control the topology of the whole spanning tree.

- Discarding: Neither forward the received frame nor learn about the source Mac address.
- Learning: Do not forward the received frame, but learn about the source Mac address, so it is a transitional status.

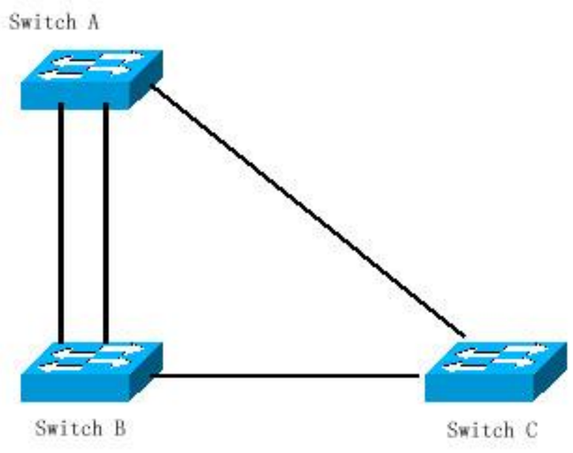
- Forwarding: Both forward the received frame and learn about the source Mac address.

For the stable network topology, only the root port and designated port can be the forwarding status, while other ports are only in the discarding status.

Generating a Network Topology Tree (Typical Application Solution)

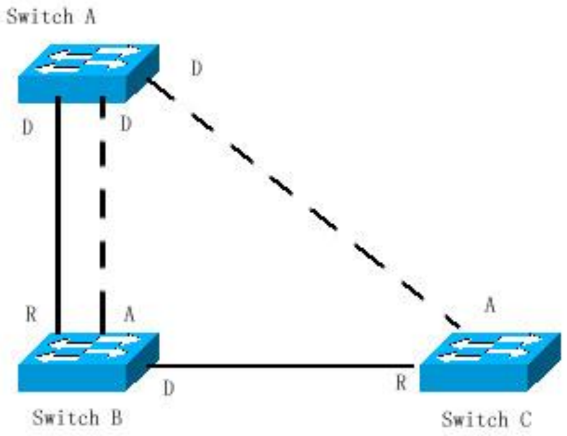
We now describe how the STP and RSTP protocols span a tree-type structure by the mixed network topology. As shown in Figure 4, the bridge IDs of Switches A, B and C are assumed in the ascending order. Namely, Switch A presents the highest priority. There is the 1000M link between switch A and switch B, and the 10M link between switch A and switch C, while it is the 100M link between switch B and switch C. Switch A acts as the backbone switch of this network and implements the link redundancy for both Switch B and Switch C. Obviously, broadcast storm would occur if all these links are active.

Figure-4



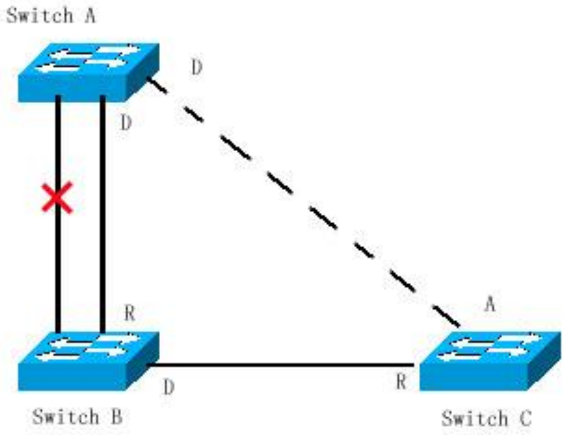
If all of these three switches enable the Spanning Tree protocol, they will select switch A as the root bridge by exchanging BPDU message. Once Switch B detects that two ports are connected to Switch A, it will select the port with the highest priority as the root port, while another one is selected as the alternate port. Meanwhile, Switch C detects that it can reach Switch A through Switch B or directly. However, Switch C discovers that the cost of the path from Switch B to Switch A is lower than that directly (For the costs corresponding to various paths, refer to table ***), so Switch C selects the port connected with Switch B as the root port, while the one that connected with Switch A as the alternate port. Various ports enter the corresponding status after their roles are determined. As a result, the network topology is generated as shown in Figure 5.

Figure-5



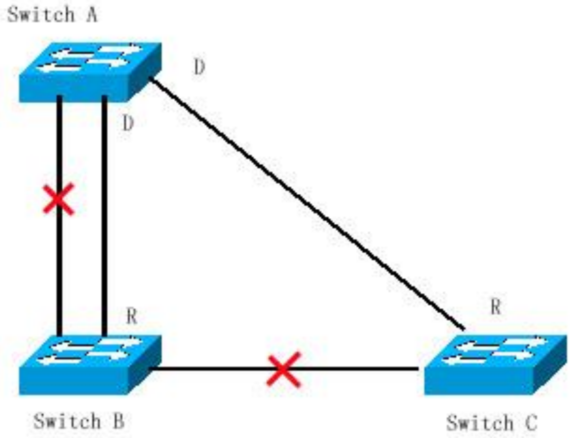
If the active path between Switch A and Switch B fails, the backup path will work. Consequently, the network topology is generated as shown in Figure 6.

Figure-6



If the path between Switch B and Switch C fails, Switch C will automatically switch the alternate port to the root port. Consequently, the network topology is generated as shown in Figure 7.

Figure-7



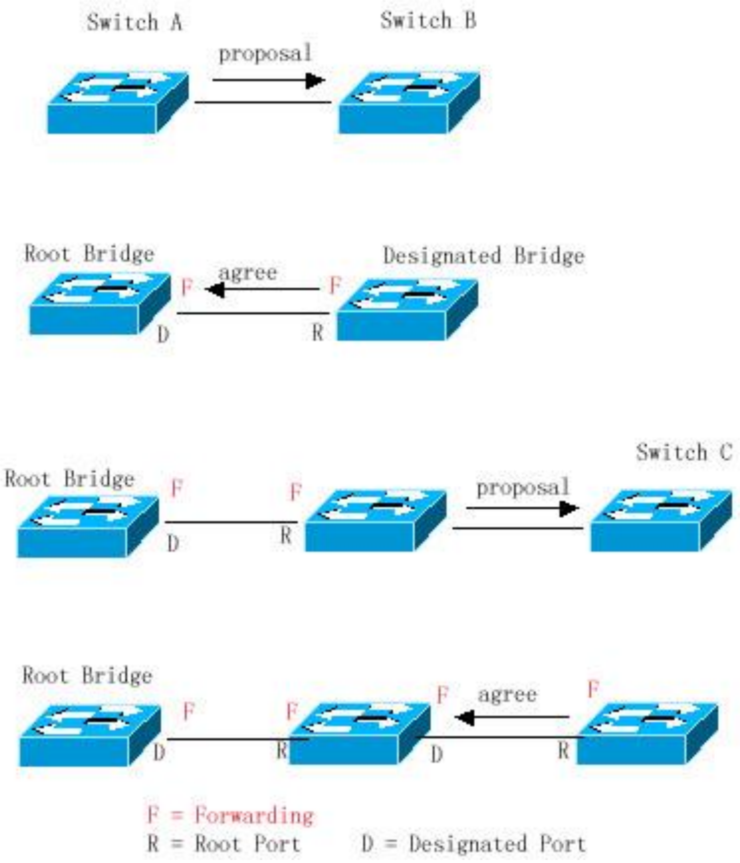
Rapid Convergence of RSTP

The following introduces the special function of RSTP: enabling rapid forwarding on a port.

The STP protocol will forward packets after 30s since the port roles are selected, which is twice as the Forward-Delay Time (you can set the Forward-Delay Time, which is 15s by default). Furthermore, the root port and designated port of each bridge will carry out the forwarding again after 30s, so it will take about 50s to stabilize the tree-type structure of the whole network topology.

The forwarding procedure of the RSTP protocol is different from that of the STP protocol. As shown in Figure 8, Switch A sends the specific proposal message of the RSTP protocol. Switch B detects that the priority of Switch A is higher than itself, takes the Switch A as the root bridge and the port that receives the message as the root port and forwards the proposal message. Then it sends the Agree message to Switch A through the root port. Upon the receipt of the proposal message, Switch A will forward the message through its designated port. After that, Switch sends the proposal message through the designated port to extend the spanning tree in turn. In theory, the RSTP protocol can immediately restore the tree-type network structure to implement rapid convergence when the network topology changes.

Figure-8



Caution

Point-to-point Connection” between ports is required for the above “handshaking” process. In order to make full use of you device, do not use non-point-to-point connection between devices.

Other than Figure 9, other schematics in this chapter are the point-to-point connection. The following lists the example figure of the non point-to-point connection.

Example of Non Point-to-point Connection:

Figure-9

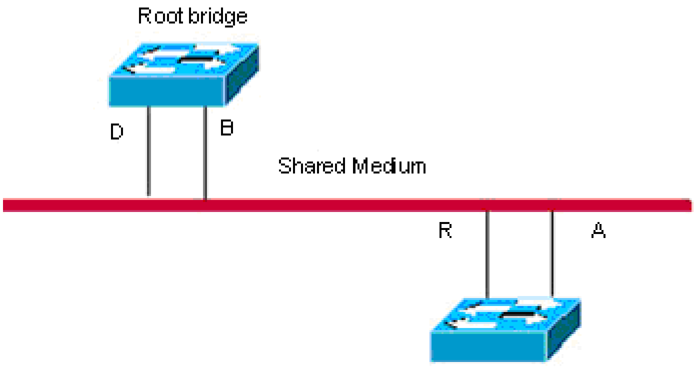
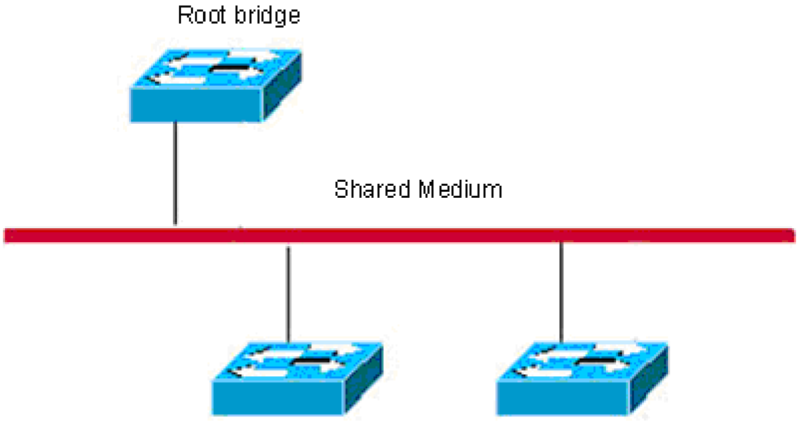
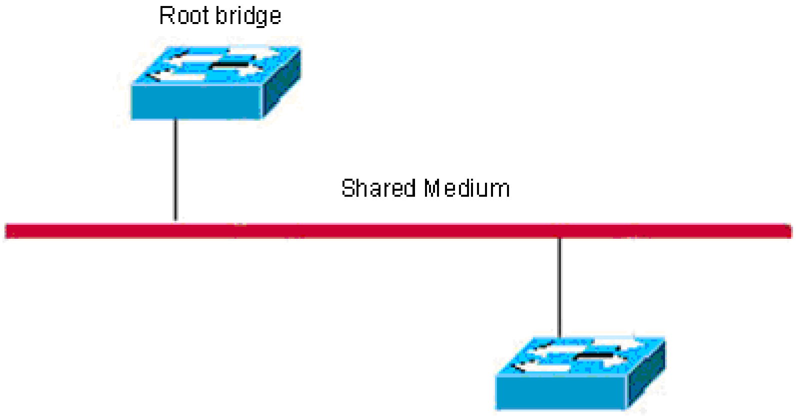


Figure-10



In addition, the following figure is a point-to-point connection and should be differentiated by users carefully.

Figure-11



Compatibility of RSTP and STP

The RSTP protocol is completely compatible with the STP protocol. It will judge whether the connected bridge supports the STP protocol or the RSTP protocol by the version number of the received BPDU message automatically. Only the forwarding process of the STP protocol is executed in the case of that the bridge supports the STP protocol. This cannot maximize the performance of the RSTP protocol.

Furthermore, using the RSTP protocol and the STP protocol will cause a problem. As shown in Figure 17-12, Switch A supports the RSTP protocol, while Switch B supports the STP protocol. Both switches are connected with each other. Switch A will send the STP BPDU message to Switch B for compatibility. However, if Switch A is connected with the RSTP-enabled Switch C, Switch A still sends the STP BPDU message, and thus causing that Switch C considers Switch A a STP-enabled bridge. As a result, two RSTP-supported switches run the STP protocol, reducing their efficiency greatly.

For this reason, the RSTP protocol provides the protocol-migration function to send the RSTP BPDU message forcibly in case that the peer bridge must support RSTP. In this way, Switch C will detect the bridge connected with it supports the RSTP protocol, so both two devices can run the RSTP protocol as shown in Figure 13.

Figure-12 Protocol Migration

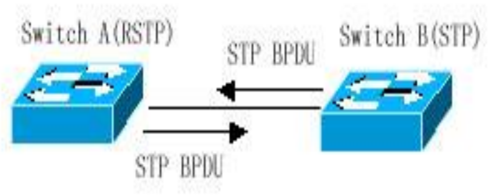
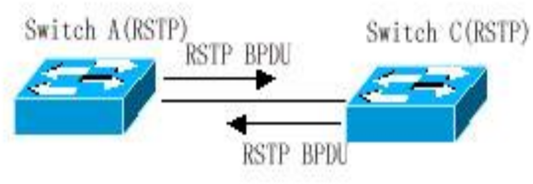


Figure-13



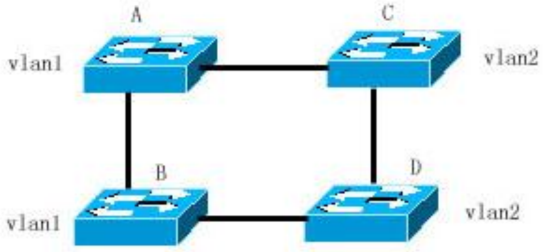
MSTP Overview

Ruijie series supports the MSTP protocol, a new spanning-tree protocol derived from the traditional STP and RSTP protocols that includes the rapid forwarding mechanism of the RSTP protocol itself.

Since traditional spanning tree protocols are not related to a VLAN, the following problems may occur in a specific network topology.

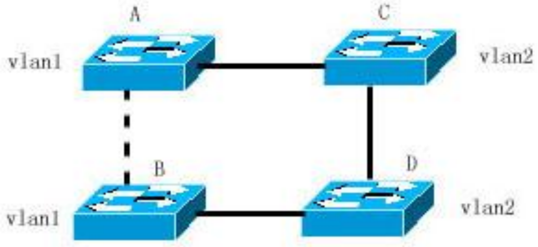
As shown in Figure 14, Switches A and B are located in Vlan1, and switches C and D in Vlan2. They form a loop.

Figure-14



If the cost of the path from Switch A through Switch C, Switch D to Switch B is smaller than that of the direct path from Switch A to Switch B, the latter path will be torn down, as shown in Figure 15. Packets in Vlan1 can not be forwarded because Switches C and D do not contain Vlan1. In this way, Vlan1 of Switch A cannot communicate with Vlan1 of Switch B.

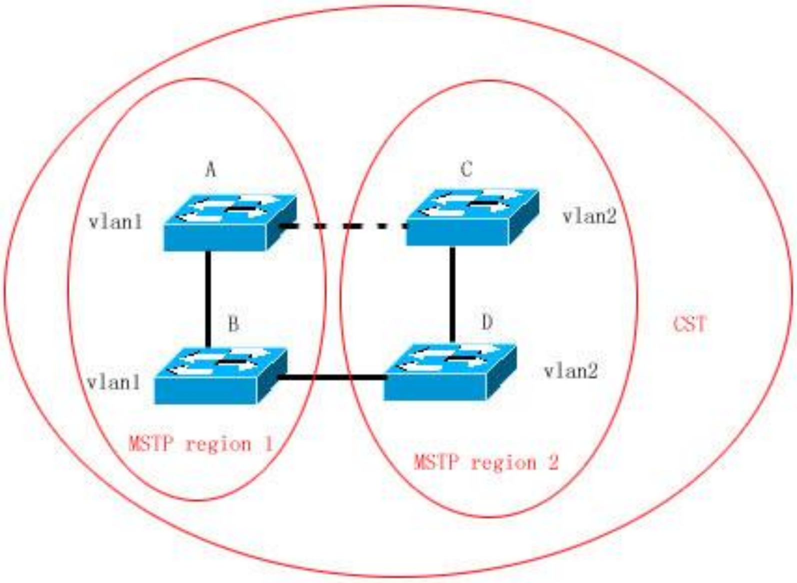
Figure-15



The MSTP protocol is developed to address this problem. It partitions one or more vlans of the switch into an instance, so the switches with the same instance configuration form a region (MST region) to run a separated spanning tree (this internal spanning-tree is referred to as the IST). The MST region is equivalent to a large device, which executes the spanning tree algorithm with other MST regions to obtain a whole spanning tree, referred to as the common spanning tree (CST).

With this algorithm, the above mentioned network can form the topology shown in Figure 16. Switches A and B are within the MSTP region 1 without a loop, so no path is discarded. This is also the case in the MSTP region 2. Region 1 and region 2 serve as two large devices respectively. There is a loop between them, so one path is discarded according to related configuration.

Figure-16



In this way, no loop occurs and the communication between the devices in a VLAN works as well.

How to Partition MSTP regions

According to above description, MSTP regions should be partitioned rationally and the switches in a MSTP region should be configured similarly for the MSTP protocol to work properly.

The MST configuration information contains:

- MST region name (name): A string of up to 32 bytes identifying the MSTP region.
- MST revision number: A revision number of 16 bits identifying the MSTP region.
- MST instance-vlan table: Each device can create up to 64 instances with IDs ranging from 1 to 64). Instance 0 always exists, so the system totally supports 65 instances. You can allocate 1 to 4094 VLANs for different instances (0 to 64) as needed, and the unallocated VLANs belong to instance 0 by default. In this way, each MSTI (MST instance) is a VLAN group and executes the spanning tree algorithm within the MSTI according to the MSTI information of the BPDU without the effect of the CIST and other MSTIs.

You can use the **spanning-tree mst configuration** command in the global configuration mode to enter the MST configuration mode and configure above information.

The MSTP BPDU carries above information. If a device has received the same MST configuration information of the BPDU as that of itself, it considers that the device connecting to this port belong to the same MST region as itself.

You are recommended to configure the instance-vlan table while the STP protocol is disable, and then enable the MSTP protocol to ensure the stability and convergence of the network topology.

Spanning Tree within a MSTP region (IST)

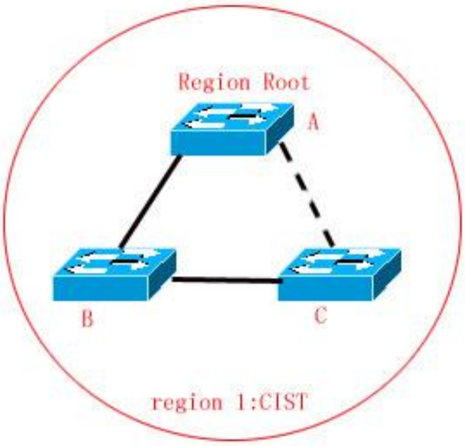
After MSTP regions are partitioned, a root bridge is elected for every instance within a region and the port role is determined for every port on a switch. A port is forwarded or discarded within an instance depneds on its role.

In this way, the IST (Internal Spanning Tree) is formed by exchanging the MSTP BPDU message, and various instances have their own spanning trees (MSTI). The spanning tree corresponding to the instance 0 is referred to as the CIST (Common Instance Spanning Tree) in conjunection with CST. That is to say, each instance provides each VLAN group with a single network topology without loop.

As shown in the following figure, Switches A, B and C form a loop within the region 1.

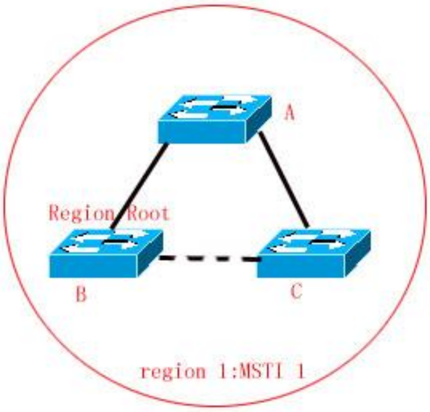
Switch A with the highest priority is selected as the region root in the CIST (instance 0). Then, the path between Switches A and C is discarded according to other parameters. Hence, for the VLAN group of instance 0, only the path from switch A to B and switch B to switch C are available, which break the loop of the VLAN group.

Figure-17



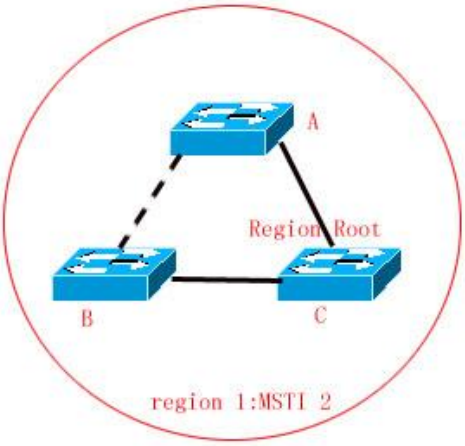
As shown in Figure 18, switch C with the highest priority is selected as the region root in the MSTI 1 (instance 1). Then, the path between switch A and B is discarded according to other parameters. Hence, for the VLAN group of instance 1, only the path from switch A to switch B and switch A to switch C are available, which break the loop of the VLAN group.

Figure-18



As shown in Figure 19, switch B with the highest priority is selected as the region root in the MSTI 2 (instance 2). Then, the path between switch B and switch C is discarded according to other parameters. Hence, for the VLAN group of instance 2, only the path from switch A to switch B and switch B to switch C are available, which break the loop of the VLAN group.

Figure-19

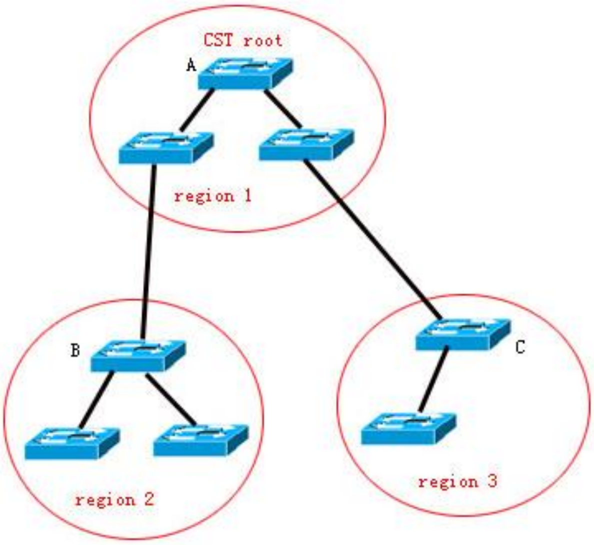


It should note that the MSTP protocol is not concerned on which VLAN a port belongs to, so users should configure corresponding path costs and priorities for ports according to actual VLAN configuration to prevent the MSTP protocol from breaking the loop unnecessarily.

Spanning Tree between MSTP regions (CST)

For CST, each MSTP region is equivalent to a large-sized device, and different MSTP regions also form a large-sized network topology tree, referred to as CST (common spanning tree). As shown in Figure 20, for CST, switch A with the smallest bridge ID is selected as the root of the entire CST (CST Root) and the CIST Regional Root in this region. In Region 2, since the root path cost from switch B to the CST root is the lowest one, switch B is selected as the CIST Regional Root in this region. Similarly, switch C is selected as the CIST Regional Root in Region 3.

Figure-20



The CIST Regional Root is not necessarily the device with the smallest bridge ID in that region. It is the device in the region that has the lowest root path cost to the CST root.

At the same time, the root port of the CIST regional root takes a new port role for the MSTI, namely the **Master port,** as the outlet of all instances, which is forwarded to all instances. In order to make the topology more stable, it is recommended to configure the outlet of the regions to the CST root on one device of this region as much as possible!

Hop Count

The IST and MSTI will not take the message age and Max age to calculate whether the BPDU message is timeout. Instead, they use the mechanism similar to the TTL of IP packets, namely hop count.

You can set it by using the **spanning-tree max-hops** command in the global configuration mode. The hop count is reduced by 1 when the BPDU message psses through a device in a region starting from the region root bridge until it is 0, which means the BPDU message is timeout. A device will discard the BPDU message whose hop count is 0.

In order to be compatible with the STP protocol and the RSTP protocol out a region, the MSTP protocol still remains the Message age and Max age mechanisms.

Compatibility of MSTP with RSTP and STP

For the STP protocol, the MSTP protocol will send the STP BPDU to be compatible with it like the RSTP protocol. For detailed information, refer to the Compatibility of RSTP and STP section.

For the RSTP protocol, it will process the CIST part of the MSTP BPDU, so it is not necessary for the MSTP to send the RSTP BPDU to be compatible with it.

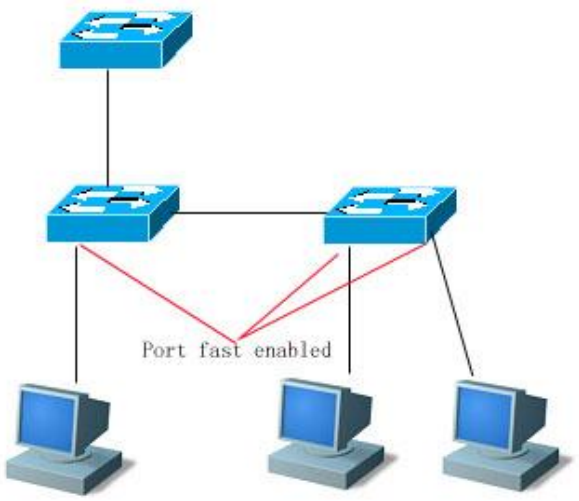
Each device that runs the STP or RSTP protocol is an independent region, and does not form the same region with any other device.

Overview of Optional Features of MSTP

Understanding Port Fast

If a port of a device is connected with the network terminal directly, this port can be set as the Port Fast to forward packets directly. The port does not need to wait 30 seconds before forwarding packets, which is the case when the port is not set to Port Fast. The following figure indicates which ports of a device can be set to Port Fast.

Figure-21



If the BPDU message is received from the Port Fast enabled port, its Port Fast operational state is disabled. At this time, this port will execute the forwarding by normal STP algorithm.

Understanding AutoEdge

If the specified port doesn't receive the BPDU message sent by the downstream port within a certain period of time (3 seconds), the port will be considered that it connects a network device and set as an edge port to enter the Forwarding status directly. An edge port will be automatically identified as a non-edge port after receiving the BPDU message.

You can disable the automatic identification function of the edge port by the **spanning-tree autoedge disabled** command.

This function is enabled by default.

Caution

When the AutoEdge function conflicts with the manually-configured Port Fast function, the latter shall prevail.

AutoEdge function can be used for rapid negotiation forwarding between the designated port and the downstream port, so the STP protocol doesn't support AutoEdge. If the designated port is in the forwarding status, Autoedge does not take effect on the port. It will take effect during repaid renegotiation such as pluging/unpluging network cables.

If a port enables the BPUD Filter, it forwards the BPDU message directly, but not be identified as the edge port automatically.

AutoEdge function is only applicable for the designated port.

AutoEdge complies with the standard definition of IEEE 802.1D (version 2004), in which the parameter range of Bridge Hello Time has been modified as 1.0-2.0. Therefore, you shall confirm that the Hello Time value is within the range when using AutoEdge function, or the risk of temporary loop will occur. It is recommended to disable AutoEdge function if it is neccesary to exceed the range of Hello Time.

Understanding BPDU Guard

The BPDU guard can be enabled globally or on individual interface. There are some slightly difference between these two ways.

You can use the **spanning-tree portfast bpduguard default** command to open the global BPDU guard enabled status in the global configuration mode. In this status, if the BPDU message is received through a Port Fast-enabled port or a AutoEdge port, this port will enter the error-disabled status, indicating the configuration error. At the same time, the port will be closed to show that some illegal users may add a network device to the network, which change the network topology.

You can also use the **spanning-tree bpduguard enable** command to enable BPDU guard on individual interface in the interface configuration mode (it is not related to whether it is AutoEdge port or not). Under this situation, it will enter the error-disabled status if this interface receives the BPDU message.

Understanding BPDU Filter

The BPDU filter can be enabled globally or on individual interface. There are some slightly difference between these two ways.

You can use the **spanning-tree portfast bpdufilter default** command to enable the BPDU filter globally in the global configuration mode. In this status, the BPDU messages can not be received or sent through a Port Fast-enabled port or a AutoEdge port, leading to no BPDU messages received by the host directly connecting the port. The BPDU filter will be disabled when the Port Fast is disabled for the AutoEdge port receives the BPDU message.

You can also use the **spanning-tree bpdufilter enable** command to enable the BPDU filter on individual interface in the interface configuration mode (it is not related to whether it is AutoEdge port or not). In this situation, this interface will not receive or transmit the BPDU message, but execute the forwarding directly.

Understanding Tc-protection

TC-BPDU messages are BPDU messages carrying with TC flag. When the L2 switch receives these messages, the network topology will change and the MAC address table will be deleted. And for L3 switch, the route table will be deleted and the port state in the ARP entry will change. To prevent the switch from processing abovementioned operations when pseudo TC-BPDU messages attack maliciously, too-heavy burden and network turbulance, the TC-protection function comes into being.

Tc-protection can only be enabled or disabled globally. It is enabled by default.

Once Tc-protection is enabled, the switch will delete the message within a certain period of time (usually 4 seconds) after receiving the TC-BPDU message while monitoring the TC-BPDU message. If it receives the TC-BPDU message during this period, it will perform the delete operation again after this period expires. This eliminates the need of frequently deleting MAC address entries and ARP entries.

Understanding TC Guard

The Tc-Protection function can reduce the removal of MAC address entries and ARP entries when a lot number of TC messages are generated in a network. However, you need to do more delete oeprations in case of TC message attack. Furthermore, the TC message is propagated and will have an effect on the whole network. The TC Guard function allows you to disable the propagation of the TC message globally or on ports. When TC Guard function is configured globally or on a port, the port will shield the TC messages received or produced to prevent from propagating them to other ports. In this way, this funciton can manage TC message attack in the network and maintain the network stability. Moreover, this function can prevent from interrupting core routes due to the oscillation of the devices on the access layer.

Caution

Network communication will be broken off if you use tc-guard function incorrectly.

You are recommended to enable this function when you ensure that there is illegal tc message attack in the network.

If you enable global tc-guard, then all the ports will not spread tc message. It is applicable for those devices that are accessed on the desk to enable this function.

If you enable interface tc-guard, then the topology change and tc message received on this port will not be spreaded to other ports. It is applicable for up-link ports especially aggregated ports to enable this function.

Understanding TC Filtering

With the TC Guard function enabled, the port will not propagate TC message to other ports participating in the spanning tree calculation on the local device. The TC message here includes the TC message received on the port, and the TC mesasge produced by the port itself. The latter one refers to the TC message generated when the forwarding state of the port changes (For example, port state change from block to forwarding), which indicates the topology may be changed.

As TC message propagation is prevented by TC Guard, the device will not clear the MAC addresses of the coppresponding ports when the topology changes, resulting in data forwarding failure.

TC filtering is introduced to solve the above problems. TC filtering will process the TC message in the condition of normal topology change instead of the TC message received on the port, so that address clearing and core route interruption caused by frequent UP/DOWN on the port without Portfast configured can be solved, and the core routing entries can be updated in time when the topology changes.

Caution

By default, the TC filtering function is disabled.

Understanding BPDU Source MAC Check

The gobal of the BPDU source MAC check funciton is to prevent malicious attack on the switch by sending the BPDU message manually and thus cause the MSTP protocol work abnormally. When the peer switch connected to a port in the point-to-point mode is determined, enabling the BPDU source MAC check function can receive only the BPDU message from the remote switch and discard all other BPDU messages to protect against malicious attacks. You can configure the corresponding MAC addresses for the BPDU source MAC check fucntion on a specific port in the interface mode. Only one MAC address is configured for one port. BPDU source MAC check can be disabled by using the **no bpdu src-mac-check** command. In this case, any BPDU message is received on the port.

Understanding Invalid Length Filtering for BPDU

When the Ethernet length field of the BPDU message exceeds 1500 bits, this BPDU message is discarded in order to avoid receiving invalid BPDU messages.

Understanding ROOT Guard

In network design, root bridge and backup root bridge are always divided in the same region. Due to error configuration of accendant and malicious attack in the network, it is possible that root bridge receives configuration message of higher priority and loses the current root bridge position, leading to error turbulance of network topology, which Root Guard function can prevent from occuring.

When enabling Root Guard, it enforces the port role of all the instances as specified port. Once the port receives configuration message of higher priority, Root Guard will set the interface as root-inconsistent (blocked). If there is no configuration message of higher priority during the time long enough, the port will be restored to be the original normal status.

You shall disable ROOT Guard function if this function results in the blocked status for interfaces and it needs manual configuration to restore to the normal status. You can use the command **spanning-tree guard none** in the interface configuration mode to disable Root Guard function. This function is enabled by default.

Caution

Incorrectly using ROOT Guard leads to network link breakdown.
If you enable ROOT Guard on non-designated port, the non-designated port will be enforced as designated port and show BKN status (blocking status).
If MST0 enters BKN status because it receives configuration message of higher priority on a port, ROOT Guard will enforce the port in all the other instances to enter BKN status.
ROOT Guard or LOOP Guard takes effect at the same time. That is , they cannot both take effect at the same time .
The AutoEdge function is disabled when enabling the ROOT Guard-enabled port.

Understanding LOOP Guard

Due to breakdown of one-way link, root port or backup port becomes designated port, being ready to forward because they can not receive BPDU, causing the loop in the network, which Loop Guard function can prevent.

For the ports configured loop guard, if they can not receive BPDU, the port roles will be migrated. However, the port state is always set as discarding till the port receive BPDU again and recalculate spanning tree.

Caution

You can enable LOOP Guard based on global or interface.
ROOT Guard or LOOP Guard takes effect at the same time. That is , they can not both take effect at the same time .
The AutoEdge function on all interfaces is ineffective when enabling LOOP Guard function globally.
The AutoEdge function on the interface is ineffective when enabling LOOP Guard function in the interface configuration mode.

Configuring MSTP

Default Spanning Tree Configuration

The following table lists the default configuration of the Spanning Tree protocol.

Item	Default value
Enable State	Disable
STP MODE	MSTP
STP Priority	32768
STP port Priority	128
STP port cost	Automatically determine according to port rate.
Hello Time	2 seconds
Forward-delay Time	15 seconds
Max-age Time	20 seconds
Default calculation method of the Path Cost	Long
Tx-Hold-Count	3
Link-type	Automatically determine by the duplex status of the port.
Maximum hop count	20
Corresponding relationship between vlan and instance	All VLANs belong to instance 0 Only instance 0 exists

You can restore the STP parameters to its default configuration (except for disabling STP) by using the **spanning-tree reset** command.

Enabling and Disabling the Spanning Tree Protocol

The spanning tree protocol is disabled on the device by default.

To enable the spanning tree protocol, execute the following command in the privileged EXEC mode:

Command	Function
Ruijie# **configure terminal**	Enter the global configuration mode.
Ruijie(config)# **spanning-tree**	Enable the spanning tree protocol.
Ruijie(config)# **end**	Return to the privileged EXEC mode.
Ruijie# **show spanning-tree**	Verify the configuration.
Ruijie# **copy running-config startup-config**	Save the configuration.

To disable the spanning tree protocol, use the **no spanning-tree** command in the global configuration mode.

Configuring the Spanning Tree Mode

According to the 802.1-related protocols, it is not necessary for administrators to set much for three versions of the spanning tree protocols such as the STP, RSTP and MSTP. These versions are compatible with one another naturally. However, given that some manufacturers will not develop the spanning tree protocol by standards, it may cause some compatibility problem. Hence, we provide a command to facilitate administrators to switch to the lower version of the spanning tree protocol for compatibility when they detect that this device is not compatible with that of other manufacturers.

Note: When you switch to the RSTP or STP version from the MSTP version, all information about MSTP Region will be cleared.

The default mode of the device is MSTP.

To enable the spanning tree protocol, execute the following command in the privileged EXEC mode:

Command	Function
Ruijie# **configure terminal**	Enter the global configuration mode.
Ruijie(config)# **spanning-tree mode** [**stp** | **rstp** | **mstp**]	Set the spanning tree version.
Ruijie(config)# **end**	Return to the privileged EXEC mode.
Ruijie# **show spanning-tree**	Verify the configuration.
Ruijie# **copy running-config startup-config**	Save the configuration.

To restore the spanning tree mode to the default value, use the **no spanning-tree mode** command in the global configuration mode.

Configuring Switch Priority

Switch priority allows you to select the root and draw the topology of a network. It is recommended that administrators set the core device with higher priority (or smaller value) to facilitate the stablization of the whole network. You can assign different switch priorities for various instances so that various instances can run separate spanning tree protocol.Only the priority of CIST (Instance 0) is related to the devices between different regions.

As mentioned in Bridge ID, there are 16 values for the priority, and all of them are multiples of 4096, which are 0, 4096, 8192, 12288, 16384, 20480, 24576, 28672, 32768, 36864, 40960, 45056, 49152, 53248, 57344, and 61440. The default value is 32768.

To configure switch priority, execute the following command in the global configuration mode:

Command	Function
Ruijie# **configure terminal**	Enter the global configuration mode.
Ruijie(config)# **spanning-tree** [**mst** *instance-id*] **priority** *priority*	Configure different switch priorities for different instances. This command configures the switch priority for instance 0 without the instance-id parameter. *instance-id*: ID of the instance in the range from 0 to 64. *priority*: switch priority in the range from 0 to 61440 and is increased by the integral multiple of 4096, 32768 by default.
Ruijie(config)# **end**	Return to the privileged EXEC mode.
Ruijie# **show running-config**	Verify the configuration.
Ruijie# **copy running-config startup-config**	Save the configuration.

To restore the switch priority to the default value, use the **no spanning-tree mst** *instance-id* **priority** command in the global configuration mode.

Configuring Port Priority

When two ports are connected to the shared media, the device will set the one of the higher priority (or smaller value) to be the forwarding status and the one of the lower priority (or larger value) to be the discarding status.If the two ports are of the same priority, the device will set the one with the smaller port number to the forwarding status. You can assign different port priorities to various instances on one port, by which various instances can run the separated spanning tree protocols.

Same as device priority, it has 16 values, all a multiple of 16. They are 0, 16, 32, 48, 64, 80, 96, 112, 128, 144, 160, 176, 192, 208, 224, and 240 respectively. The default value is 128.

To configure a port priority, execute the following commands in the privileged EXEC mode:

Command	Function
Ruijie# **configure terminal**	Enter the global configuration mode.
Ruijie(config)# **interface** *interface-id*	Enter the interface configuration mode. A legal interface contains a physical port and an aggregate Link.
Ruijie(config-if)# **spanning-tree** [**mst** *instance-id*] **port-priority** *priority*	Configure different priorities for different instances. The command without the *instance-id* parameter will configure a port priority for instance 0. *instance-id*: Interface ID in the range from 0 to 64. *priority*: Port priority of an instance in the range from 0 to 240. Furthermore, it is increased by the integral multiple of 16, 128 by default.
Ruijie(config-if)# **end**	Return to the privileged EXEC mode.
Ruijie# **show spanning-tree** [**mst** *instance-id*] **interface** *interface-id*	Verify the configuration.
Ruijie# **copy running-config startup-config**	Save the configuration.

To restore the port priority to the default value, execute the **no spanning-tree mst** *instance-id* **port-priority** command in the interface configuration mode.

Configuring Path Cost of a Port

The switch determines a root port upon the total of the path costs along the path from a port to the boot bridge. The port the total of paths costs from the port to the root brdige is the smallest is elected the root port. Its default value is calculated by the media speed of the port automatically. The higher the media speed, the smaller the cost is. It is not necessary for administrators to change it for the path cost calculated in this way is most scientific. You can assign different cost paths for various instances on one port, by which various instances can run the separated spanning tree protocols.

To configure the path cost of a port, execute the following commands in the privileged EXEC mode:

Command	Function
Ruijie# **configure terminal**	Enter the global configuration mode.

Ruijie(config)# **interface** *interface-id*	Enter the interface configuration mode. A legal interface contains a physical port and an aggregate Link.
Ruijie(config-if)# **spanning-tree** [**mst** *instance-id*] **cost** *cost*	Configure different priorities for different instances. The command without the *instance-id* parameter will configure a port priority for instance 0. *instance-id*: Interface ID in the range of 0 to 64. *cost*: Path cost of the port in the range of 1 to 200,000,000. The default value is calculated by the media rate of the port automatically.
Ruijie(config-if)# **end**	Return to the privileged EXEC mode.
Ruijie# **show spanning-tree** [**mst** *instance-id*] **interface** *interface-id*	Verify the configuration.
Ruijie# **copy running-config startup-config**	Save the configuration.

To restore the path cost of a port to the default value, execute the **no spanning-tree mst** *cost* command in the interface configuration mode.

Configuring the Default Calculation Method of Path Cost (path cost method)

If the path cost of a port is the default value, the device will calculate the path cost of this port by port rate. However, IEEE 802.1d-1998 and IEEE 802.1t specify different path cost values for a port rate respectively. The value range of the 802.1d-1998 is short (1 to 65535), while the value range of the 802.1t is long (1 to 200,000,000).

There are two modes for the Cost value of AP: 1) our private mode fixes it to: the Cost value of the physical port * 95%; 2) the standard value is 20,000,000,000/ the actual link bandwidth of AP (The actual link bandwidth is: the bandwidth of member port * the number of UP member ports). Administrators should unify the path cost standard of the whole network. The default mode is long (IEEE 802.1t Mode).

The following table lists the path costs set for different port rates in two standards.

Port Rate	Interface	IEEE 802.1d (short)	IEEE 802.1t (long)	IEEE 802.1t (long standard)
10M	Common Port	100	2000000	2000000
	Aggregate Link	95	1900000	2000000 ÷ linkupcnt
100M	Common Port	19	200000	200000
	Aggregate Link	18	190000	200000 ÷ linkupcnt
1000M	Common Port	4	20000	20000
	Aggregate Link	3	19000	20000 ÷ linkupcnt
10000M	Common Port	2	2000	2000
	Aggregate Link	1	1900	2000 ÷ linkupcnt

Caution

1. The default path cost mode is long. After changing the path cost to the standard mode, the cost of AP will vary with the number of UP member ports. The change of port cost value may result in network topology change.
2. For the static AP, the linkupcnt in the table is the number of UP member ports; for the LACP AP, the linkupcnt refers to the number of member ports participating in AP data forwarding. If there is no linkup on the member port, the linkupcnt is 1. For detailed configurations about AP and LACP, refer to *AP-SCG* and *LACP-SCG*.

To configure the default calculation method of path cost, execute the following commands in the privileged EXEC mode:

Command	Function
Ruijie# **configure terminal**	Enter the global configuration mode.
Ruijie(config)# **spanning-tree pathcost method** { { **long** [**standard**] } | **short** }	Configure the default calculation method of the port path cost as long, standard long or short, with long by default.
Ruijie(config)# **end**	Return to the privileged EXEC mode.
Ruijie# **show running-config**	Verify the configuration.
Ruijie# **copy running-config startup-config**	Save the configuration.

To restore the setting to the default value, execute the **no spanning-tree pathcost** method command in the global configuration mode.

Configuring Hello Time

Configure the interval of sending the BPDU message. The default value is 2s.

To configure the Hello Time, execute the following commands in the privileged EXEC mode:

Command	Function
Ruijie# **configure terminal**	Enter the global configuration mode.
Ruijie(config)# **spanning-tree hello-time** *seconds*	Configure the hello time ranging from 1 to 10s, 2s by default.
Ruijie(config)# **end**	Return to the privileged EXEC mode.
Ruijie# **show running-config**	Verify the configuration.
Ruijie# **copy running-config startup-config**	Save the configuration.

To restore the hello time to the default value, execute the **no spanning-tree hello-time** command in the global configuration mode.

Configuring Forward-Delay Time

Configure the interval for changing port status. The default value is 15s.

To configure the forward-delay time, execute the following commands in the global configuration mode:

Command	Function

Ruijie# **configure terminal**	Enter the global configuration mode.
Ruijie(config)# **spanning-tree forward-time** *seconds*	Configure the forward delay time ranging from 4 to 30 seconds, 15 seconds by default.
Ruijie(config)# **end**	Return to the privileged EXEC mode.
Ruijie# **show running-config**	Verify the configuration.
Ruijie# **copy running-config startup-config**	Save the configuration.

To restore the forward-delay time to the default value, execute the **no spanning-tree forward-time** command in the global configuration mode.

Configuring Max-Age Time

Configure the maximum period of time before the BPDU message is aged out. The default value is 20s.

In the privilege mode, perform these steps to configure the Max-Age Time:

Command	Function
Ruijie# **configure terminal**	Enter the global configuration mode.
Ruijie(config)# **spanning-tree max-age** *seconds*	Configure the max age time ranging from 6 to 40 seconds, 20 seconds by default.
Ruijie(config)# **end**	Return to the privileged EXEC mode.
Ruijie# **show running-config**	Verify the configuration.
Ruijie# **copy running-config startup-config**	Save the configuration.

To restore the max age time to the default value, execute the **no spanning-tree max-age** command in the global configuration mode.

Caution

Hello Time, Forward-Delay Time and Max-Age Time have their own value ranges. Meanwhile, the following condition must be addressed: 2*(Hello Time + 1.0 seconds) <= Max-Age Time <= 2*(Forward-Delay – 1.0 second). Otherwise, it may cause the topology instability.

Configuring Tx-Hold-Count

Configure the maximum number of the BPDU message sent per second, 3 by default.

To configure the Tx-Hold-Count, execute the following commands in the global configuration mode:

Command	Function
Ruijie# **configure terminal**	Enter the global configuration mode.
Ruijie(config)# **spanning-tree tx-hold-count** *numbers*	Configure the maximum number of the BPDU message sent per second in the range of 1 to10, 3 by default.
Ruijie(config)# **end**	Return to the privileged EXEC mode.
Ruijie# **show running-config**	Verify the configuration.

Ruijie# **copy running-config startup-config**	Save the configuration.

To restore the default setting, execute the **no spanning-tree tx-hold-count** command in the global configuration mode.

Configuring Link-type

Configure the link-type of a port. This is crucial for rapid RSTP convergence. For details, refer to Rapid RSTP Convergence. Without configuration, the device will set the link type of a port according to its duplex status automatically, with point-to-point for the full duplex port and shared for the half duplex port.

To configure the link type of a port, execute the following commands in the interface configuration mode:

Command	Function
Ruijie# **configure terminal**	Enter the global configuration mode.
Ruijie(config)# **interface** *interface-id*	Enter the interface configuration mode.
Ruijie(config-if)# **spanning-tree link-type point-to-point**/**shared**	Configure the link type of the interface, with point-to-point for the full duplex port and shared for the half duplex port. Point-to-point indicates the rapid forwarding is enabled on the port.
Ruijie(config-if)# **end**	Return to the privileged EXEC mode.
Ruijie# **show running-config**	Verify the configuration.
Ruijie# **copy running-config startup-config**	Save the configuration.

To restore the link type of a port to the default value, execute the **no spanning-tree link-type** command in the interface configuration mode.

Configuring Protocol Migration Processing

This command is to check the version globally or on individual port. For related information, refer to Compatibility of RSTP and STP.

Command	Function
Ruijie# **clear spanning-tree detected-protocols**	Forcibly check the version on all ports.
Ruijie# **clear spanning-tree detected-protocols interface** *interface-id*	Check the version forcibly on the port.

Configuring an MSTP Region

To deploy several devices in the same MSTP Region, you have to configure these devices with the same name, the same revision number, and the same Instance-VLAN table.

You can assign a VLAN to instances 0 to 64 respectively as required. The remaining VLANs will be automatically assigned to instance 0. One vlan can only be of an instance.

It is recommended to configure the Instance-VLAN table when the MSTP protocol is disabled. After configuration, you should enable the MSTP protocol again to ensure the stability and convergence of the network topology.

To configure an MSTP region, execute the following commands in the global configuration mode:

Command	Function
Ruijie# **configure terminal**	Enter the global configuration mode.
Ruijie(config)# **spanning-tree mst configuration**	Enter the MST configuration mode.
Ruijie(config-mst)# **instance** *instance-id* **vlan** *vlan-range*	Add a VLAN group to a MST instance. *instance-id*: Instance ID ranging from 0 to 64. *vlan-range:* VLAN range in the range 1 to 4094. For instance: The **instance 1 vlan 2-200** command is to add VLAN 2-200 to instance 1. The **instance 1 vlan 2,20,200** command is to add VLAN 2, VLAN 20 and VLAN 200 to instance 1. You can use the **no** option of this command to delete a VLAN from an instance, and the deleted VLAN will be added to instance 0 automatically.
Ruijie(config-mst)# **name** *name*	Specify the MST configuration name, a string of up to 32 bytes.
Ruijie(config-mst)# **revision** *version*	Specify the MST revision number in the range 0 to 65535. The default value is 0.
Ruijie(config-mst)# **show spanning-tree mst configuration**	Verify the configuration.
Ruijie(config-mst)# **end**	Return to the privileged EXEC mode.
Ruijie# **copy running-config startup-config**	Save the configuration.

To restore the MST region configuration to the default value, execute the **no spanning-tree mst configuration** command in the global configuration mode. You can use the **no instance** *instance-id* command to delete an instance. Similarly, the **no name** and **no revision** commands can be used to restore the MST name and MST revision number settings to the default value, respectively.

The following is the example of configuration:

```
Ruijie(config)# spanning-tree mst configuration
Ruijie(config-mst)# instance 1 vlan 10-20
Ruijie(config-mst)# name region1
Ruijie(config-mst)# revision 1
Ruijie(config-mst)# show spanning-tree mst configuration
Multi spanning tree protocol : Enable Name [region1]
Revision 1
Instance Vlans Mapped
-------- ---------------------
0 1-9,21-4094
```

```
1 10-20
-------------------------------
Ruijie(config-mst)# exit
Ruijie(config)#
```

Caution

Before configuring vlan and instance mapping relationship, please ensure that all configured VLANs have been created. Otherwise, the association of vlan and instance on part of the products may be failed.

Configuring Maximum-Hop Count

Maximum-Hop Count means how many devices the BPDU message will pass through in a MSTP region before being discarded. This parameter takes effect for all instances.

To configure the Maximum-Hop Count, execute the following commands in the global configuraiton mode:

Command	Function
Ruijie# **configure terminal**	Enter the global configuration mode.
Ruijie(config)# **spanning-tree max-hops** *hop-count*	Configure the Maximum-Hop Count ranging from 1 to 40, 20 by default.
Ruijie(config)# **end**	Return to the privileged EXEC mode.
Ruijie# **show running-config**	Verify the configuration.
Ruijie# **copy running-config startup-config**	Save the configuration.

To restore the Maxium-Hop Count to the default value, execute the **no spanning-tree max-hops** command in the global configuration mode.

Configuring Intereface Compatibility Mode

In interface compatibility mode, when a port sends BPDU, it will carry different MSTI information according to the current port attribute to realize interconnection with other vendors.

To configure the interface compatibility mode, execute the following commands in the interface configuration mode:

Command	Function
Ruijie# **configure terminal**	Enter the global configuration mode.
Ruijie(config)#**interface** *interface-id*	Enter the Interface configuration mode.
Ruijie(config-if)# **spanning-tree compatible enable**	Enable interface compatibility mode.
Ruijie(config-if)# **end**	Return to the privileged EXEC mode.
Ruijie# **show running-config**	Check configuration items.
Ruijie# **copy running-config startup-config**	Save the configuration.

To remove the settings, you can execute command **no spanning-tree compatible enable.**

Configuring Optional MSTP Features

Default Setting of Optional Spanning Tree Features

All the optional features are disabled by default, except for AutoEdge function.

Enabling Port Fast

Enabling Port Fast lets a port directly forward the BPDU message. When Port Fast is disabled due to the receipt of the BPDU message, the port will participate in the STP algorithm and forward the BPDU message normally.

To enable Port Fast, execute the following commands in the global configuraiton mode:

Command	Function
Ruijie# **configure terminal**	Enter the global configuration mode.
Ruijie(config)# **interface** *interface-id*	Enter the interface configuration mode. A legal interface contains a physical port and an Aggregate Link.
Ruijie(config-if)# **spanning-tree Portfast**	Enable Port Fast on the interface.
Ruijie(config-if)# **end**	Return to the privileged EXEC mode.
Ruijie# **show spanning-tree interface** *interface-id* **portfast**	Verify the configuration.
Ruijie# **copy running-config startup-config**	Save the configuration.

To disable Port Fast, execute the **spanning-tree portfast disable** command in the interface configuration mode.

You can use the **spanning-tree portfast default** command in the global configuration mode to enable Port Fast on all ports.

Enabling BPDU Guard

After BPDU Guard is enabled, a port will in the error-disabled status after receiving the BPDU packet.

To configure the BPDU guard, execute the following commands in the global configuration mode:

Command	Function
Ruijie# **configure terminal**	Enter the global configuration mode.
Ruijie(config)# **spanning-tree portfast Bpduguard default**	Enable the BPDU Guard globally.
Ruijie(config)# **interface** *interface-id*	Enter the interface configuration mode. A legal interface contains a physical port and an aggregate link.
Ruijie(config-if)# **spanning-tree portfast**	Enable Port Fast on the interface before the bpduguard configuration takes effect globally.
Ruijie(config-if)# **end**	Return to the privileged EXEC mode.
Ruijie# **show running-config**	Verify the configuration.

Ruijie# **copy running-config startup-config**	Save the configuration.

To disable BPDU Guard, execute the **no spanning-tree portfast bpduguard default** command in the global configuration command.

To enable or disable BPDU Guard on an interface, execute the **spanning-tree bpduguard enable** command or the **spanning-tree bpduguard disable** command on the interface respectively.

Enabling BPDU Filter

A port neither transmit nor receive the BPDU message after the BPDU filter is enabled.

To configure the BPDU Filter, execute the following commands in the global configuration mode:

Command	Function
Ruijie# **configure terminal**	Enter the global configuration mode.
Ruijie(config)# **spanning-tree portfast bpdufilter default**	Enable BPDU filter globally.
Ruijie(config)# **interface** *Interface-id*	Enter the interface configuration mode. A legal interface contains a physical port and an aggregate link.
Ruijie(config-if)# **spanning-tree Portfast**	Enable portfast on this interface before the bpduguard configuration takes effect globally.
Ruijie(config-if)# **end**	Return to the privileged EXEC mode.
Ruijie# **show running-config**	Verify the configuration.
Ruijie# **copy running-config startup-config**	Save the configuration.

To restore the default setting, execute the **no spanning-tree portfast bpdufilter default** command in the global configuration mode.

To enable or disable BPDU Filter on an interface, execute the **spanning-tree bpdufilter enable** command or the **spanning-tree bpdufilter disable** command in the interface configuration mode.

Enabling Tc_Protection

To configure Tc_Protection, execute the following commands in the global configuration mode:

Command	Function
Ruijie# **configure terminal**	Enter the global configuration mode.
Ruijie(config)# **spanning-tree tc-protection**	Enable Tc-Protection
Ruijie(config)# **end**	Return to the privileged EXEC mode.
Ruijie# **show running-config**	Verify the configuration.
Ruijie# **copy running-config startup-config**	Save the configuration.

To disable Tc_Protection, execute the **no spanning-tree tc-protection** command in the global configuration mode.

Enabling TC Guard

To enable TC Guard globally, execute the following commands in the global configuration mode:

Command	Function
Ruijie# **configure terminal**	Enter the global configuration mode.
Ruijie(config)# **spanning-tree tc-protection tc-guard**	Enable TC Guard globally.
Ruijie(config)# **end**	Return to the privileged EXEC mode.
Ruijie# **show running-config**	Verify the configuration.
Ruijie# **copy running-config startup-config**	Save the configuration.

To configure TC Guard on an interface, execute the following commands in the interface configuration mode:

Command	Function
Ruijie# **configure terminal**	Enter the global configuration mode.
Ruijie(config)# **interface** *Interface-id*	Enter the interface configuration mode. A legal interface includes a physical port and an aggregate link.
Ruijie(config-if)# **spanning-tree tc-guard**	Enable TC Guard on this interface.
Ruijie(config-if)# **end**	Return to the privileged EXEC mode.
Ruijie# **show running-config**	Verify the configuration.
Ruijie# **copy running-config startup-config**	Save the configuration.

Enabling TC Filtering

To enable TC filtering, execute the following commands:

Command	Function
Ruijie# **configure terminal**	Enter the global configuration mode.
Ruijie(config)# **interface** *interface-id*	Enter interface configuration mode. The valid interface includes physical port and Aggregate Link.
Ruijie(config)# **spanning-tree ignore tc**	Enable TC filtering for this interface.
Ruijie(config)# **end**	Return to the privileged EXEC mode.
Ruijie# **show running-config**	Verify the configuration.
Ruijie# **copy running-config startup-config**	Save the configuration.

To disable TC filtering, use the **no spanning-tree ignore tc** command in interface configuration mode.

Enabling BPDU Source MAC check

After the BPDU source MAC check is enabled, the switch accepts only the BPDU message from the specified MAC address.

To configure the BPDU source MAC check, execute the following commands in the interface configuration mode:

Command	Function
Ruijie# **configure terminal**	Enter the global configuration mode.
Ruijie(config)# **interface** *interface-id*	Enter the interface configuration mode. A legal interface includes a physical port and an aggregate link.
Ruijie(config-if)#**bpdu src-mac-check** H.H.H	Enable the BPDU source MAC address check function on the interface.
Ruijie(config-if)# **end**	Return to the privileged EXEC mode.
Ruijie# **show running-config**	Verify the configuration.
Ruijie# **copy running-config startup-config**	Save the configuration.

To disable BPDU source MAC check, execute the **no bpdu src-mac-check** command in the interface mode.

Enabing Root Guard

To configure interface ROOT Guard, execute the following commands in the privileged mode:

Command	Function
Ruijie# **configure terminal**	Enter the global configuration mode.
Ruijie(config)# **interface** *interface-id*	Enter the interface configuration mode. Valid interface includes physical port and Aggregate Link.
Ruijie(config-if)# **spanning-tree guard root**	Enable interface ROOT Guard.
Ruijie(config-if)# **end**	Return to the privileged EXEC mode.
Ruijie# **show running-config**	Verify the configuration.
Ruijie# **copy running-config startup-config**	Save the configuration.

Enabling Loop Guard

To configure global LOOP Guard, execute the following commands in the privileged EXEC mode:

Command	Function
Ruijie# **configure terminal**	Enter the global configuration mode.
Ruijie(config)# **spanning-tree loopguard default**	Enable global LOOP Guard.
Ruijie(config)# **end**	Return to the privileged EXEC mode.
Ruijie# **show running-config**	Verify the configuration.
Ruijie# **copy running-config startup-config**	Save the configuration.

To configure interface LOOP Guard, execute the following commands in the privileged EXEC mode:

Command	Function
Ruijie# **configure terminal**	Enter the global configuration mode.
Ruijie(config)# **interface** *interface-id*	Enter the interface configuration mode. Valid interface includes physical port and Aggregate Link.
Ruijie(config-if)# **spanning-tree guard loop**	Enable interface Loop Guard.

Ruijie(config-if)# **end**	Return to the privileged EXEC mode.
Ruijie# **show running-config**	Verify the configuration.
Ruijie# **copy running-config startup-config**	Save the configuration.

Disabling Interface Guard

To disable interface ROOT or LOOP Guard, execute the following commands in the privileged EXEC mode:

Command	**Function**
Ruijie# **configure terminal**	Enter the global configuration mode.
Ruijie(config)# **interface** *interface-id*	Enter the interface configuration mode. Valid interface includes physical port and Aggregate Link.
Ruijie(config-if)# **spanning-tree guard none**	Disable interface Loop Guard.
Ruijie(config-if)# **end**	Return to the privileged EXEC mode.
Ruijie# **show running-config**	Verify the configuration.
Ruijie# **copy running-config startup-config**	Save the configuration.

Showing MSTP Configuration and Status

You can use the following show commands to view the configuration of MSTP:

Command	**Meaning**
Ruijie# **show spanning-tree**	Display the information on the parameters and topology of MSTP.
Ruijie# **show spanning-tree summary**	Display the information on various instances and port forwarding status of MSTP.
Ruijie# **show spanning-tree inconsistentports**	Display the block port due to root guard or loop guard.
Ruijie# **show spanning-tree mst configuration**	Display the configuration information of the MST region.
Ruijie# **show spanning-tree mst** *instance-id*	Display the MSTP information of an instance.
Ruijie# **show spanning-tree mst** *instance-id* **interface** *interface-id*	Display the MSTP information of the specified instance of the interface.
show spanning-tree mst *instance-id* **topochange record**	Display the STP topology change record.
Ruijie# **show spanning-tree interface** *interface-id*	Display the MSTP information of all the instances of the interface.
Ruijie# **show spanning-tree forward-time**	Display forward-time.
Ruijie# **show spanning-tree Hello Time**	Display Hello time.
Ruijie# **show spanning-tree max-hops**	Display max-hops.

Ruijie# **show spanning-tree tx-hold-count**	Display tx-hold-count.
Ruijie# **show spanning-tree pathcost Method**	Display pathcost method.

Configuring LLDP

LLDP Overview

Drafted by IEEE 802.1AB, LLDP (Link Layer Discovery Protocol) can detect network topology change and identify what the change is. With LLDP, a device sends local device information as TLV (Type, Length and Value) triplets in LLDP Data Units (LLDPDUs) to the neighbor devices, and at the same time, stores the device information received in LLDPDUs sent from the LLDP neighbors in a standard management information base (MIB) to be accessed by the network management system.

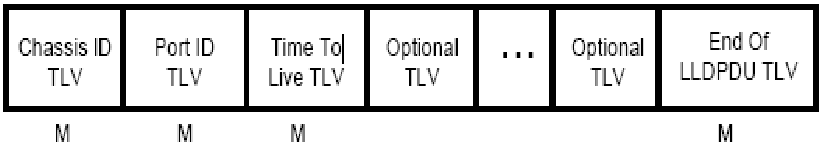
Through LLDP, the network management system can learn about the state of topological connections, such as which ports of the device are connected to other devices, the rate of ports on both sides of link, and whether the duplex mode is matched. The network administrator can quickly locate and eliminate faults according to such information.

Basic Concepts

LLDPDU

LLDPDU refers to the data units encapsulated in LLDP packets, and comprises multiple TLV sequences, including three fixed TLVs, a number of optional TLVs and an End of TLV. The detailed format of LLDPDU is shown in Fig 1:

Fig 1-1 LLDPDU format



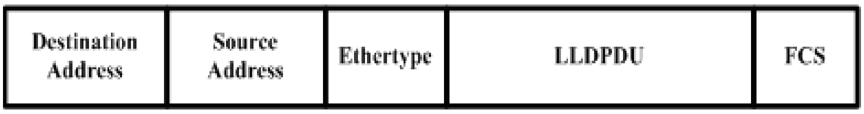
- * M refers to fixed TLV.
- In LLDPDU, Chassis ID TLV, Port ID TLV, Time To Live TLV and End Of LLDPDU TLV are fixed TLVs, while other TLVs are optional.

LLDPDU Encapsulation Format

LLDP packet supports two encapsulation formats: Ethernet II and SNAP (Subnetwork Access Protocols).

Ethernet II encapsulated LLDPDU format is shown in Fig 2:

Figure 1-2 Ethernet II encapsulated LLDPDU format



Specifically:

- Destination Address: destination MAC address. It is fixed to 01-80-C2-00-00-0E, a multicast address.

- Source Address: source MAC address, layer-2 MAC address of device.

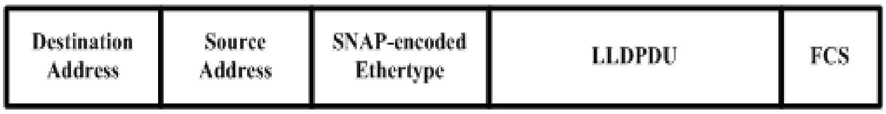
- Ethertype: the Ethernet type, 0x88CC.

- LLDPDU: LLDP Data Unit.

- FCS: frame check sequence.

SNAP-encapsulated LLDPDU format is shown in Fig 3:

Figure 1-3 SNAP-encapsulated LLDPDU format



Destination Address	Source Address	SNAP-encoded Ethertype	LLDPDU	FCS

Specifically:

- Destination Address: destination MAC address. It is fixed to 01-80-C2-00-00-0E, a multicast address.

- Source Address: source MAC address, layer-2 MAC address of device.

- SNAP-encoded Ethertype: SNAP-encapsulated Ethernet type, AA-AA-03-00-00-00-88-CC.

- LLDPDU: LLDP Data Unit.

- FCS: frame check sequence.

TLV

TLVs encapsulated in LLDPDU can fall into two broad categories:

- Basic management TLVs

- Organizationally specific TLVs

Basic management TLVs are a group of basic TLVs for network management. The organizationally specific TLVs are TLVs defined by standards organizations and other organizations, such as IEEE 802.1, IEEE 802.3 and etc.

1) Basic management TLVs

Basic management TLVs include two types of TLVs: fixed TLVs and optional TLVs. Fixed TLVs must be included in LLDPDU, while optional TLVs can be included or excluded according to need.

Basic management TLVs are shown in Table 1:

Type	Description	Use in LLDPDU
End Of LLDPDU TLV	End mark of LLDPDU, occupying 2 bytes	Fixed
Chassis ID TLV	Used to identify the device, and is generally represented with MAC address	Fixed
Port ID TLV	ID of the LLDPDU sending port	Fixed
Time To Live TLV	Life of local information on the neighbor device. When TLV with 0 TTL is received, the corresponding neighbor information must be deleted.	Fixed
Port Description TLV	Port description of LLDPDU sending port	Optional
System Name TLV	Name of the sending device	Optional
System Description TLV	Description of the sending device, including hardware/software version, operating system and etc.	Optional
System Capabilities TLV	Identifies the primary functions of the sending device, such as bridging, routing and relaying.	Optional
Management Address TLV	Management address, including interface number and OID (object Identifier).	Optional

Table 1 Basic management TLV

☑ Basic management TLVs are supported by the LLDP protocol used by Ruijie switch products.

2) Organizationally specific TLVs

Different organizations (such as IEEE 802.1, IEEE 802.3, IETF or device suppliers) may define specific TLVs to advertise specific information about the device, and OUI (Organizationally Unique Identifier) is used to identify different organizations.

Organizationally specific TLVs are optional TLVs advertised in LLDPDU according to user's actual needs. Currently, commonly found organizationally specific TLVs include:

1) IEEE 802.1 organizationally specific TLVs

IEEE 802.1 organizationally specific TLVs are shown in Table 2:

Type	Description
Port VLAN ID TLV	VLAN identifier of the sending port
Port And Protocol VLAN ID TLV	Protocol VLAN identifier of the sending port
VLAN Name TLV	Name of VLAN with which the device is configured
Protocol Identity TLV	Protocols supported by the port

Table 2 IEEE 802.1 organizationally specific TLVs

☑ LLDP protocol used by Ruijie switch products doesn't support the sending of Protocol Identity TLV, but allows the reception of such TLV.

2) IEEE 802.3 organizationally specific TLVs

IEEE 802.3 organizationally specific TLVs are shown in Table 3:

Type	Description
MAC/PHY Configuration/Status TLV	The bit-rate and duplex capabilities of the sending port and support for auto negotiation.
Power Via MDI TLV	Power supply capability of the port
Link Aggregation TLV	Indicate the link aggregation capability of the port and the aggregation status.
Maximum Frame Size TLV	The maximum frame size supported by the port.

Table 3 IEEE 802.3 organizationally specific TLVs

☑ IEEE 802.3 organizationally specific TLVs are supported by the LLDP protocol used by Ruijie switch products.

3) LLDP-MED TLVs

LLDP-MED is the extension of IEEE 802.1AB LLDP protocol, so that the user can conveniently deploy VoIP (Voice Over IP) network and fault detection. It provides multiple applications such as network policy configuration, device detection, PoE management and directory management, providing a cost-effective and easy-to-use solution for deploying voice devices in Ethernet.

LLDP-MED TLVs are shown in Table 4:

Type	**Description**
LLDP-MED Capabilities TLV	Whether the device supports LLDP-MED, the type of LLDP-MED TLV encapsulated in LLDPDU, and the type of current device (network connection device or endpoint)
Network Policy TLV	Advertise VLAN configuration of the specific port, supported applications (voice and video, for example), and the Layer 2 priorities.
Location Identification TLV	Location identifier information for an endpoint, used to accurately locate the endpoint in applications such as network topology collection.
Extended Power-via-MDI TLV	Provide more advanced power supply management.
Inventory – Hardware Revision TLV	Hardware version of MED device
Inventory – Firmware Revision TLV	Firmware version of MED device
Inventory – Software Revision TLV	Software version of MED device
Inventory – Serial Number TLV	Serial number of MED device
Inventory – Manufacturer Name TLV	Vendor name of MED device
Inventory – Model Name TLV	Model name of MED device
Inventory – Asset ID TLV	Asset ID of MED device, used for directory management and asset tracking.

Table 4 LLDP-MED TLVs

☑ LLDP-MED TLVs are supported by the LLDP protocol used by Ruijie switch products.

Working Principles

Operating Modes of LLDP

LLDP provides three operating modes:

- TxRx: sending and receiving LLDPDUs.
- Rx Only: only sending LLDPDUs.
- Tx Only: only receiving LLDPDUs.

When the LLDP operating mode of a port changes, the port will initialize the protocol state machine. To prevent LLDP from being initialized too frequently during times of frequent operating mode change, you can configure a re-initialization delay.

Mechanism for Transmitting LLDPDUs

An LLDP-enabled port operating in TxRx mode or Tx Only mode will send LLDPDUs both periodically and when the local device information changes. To avoid frequent LLDPDU sending during times of frequent local device information change, an interval is introduced between two successive LLDPDUs. This interval can be configured manually.

LLDP provides two types of packets:

- Standard LLDPDUs: including the management and configuration information about local device.
- Shutdown LLDPDU: When LLDP sending mode is disabled or when the port is administratively shut down, shutdown LLDPDU will be sent. Shutdown LLDPDU generally comprises Chassis ID TLV, Port ID TLV, Time To Live TLV and End Of LLDP TLV, with the TTL in Time To Live TLV being 0. When the device receives shutdown LLDPDUs, it will consider the neighbor on longer available and delete neighbor information.

When LLDP operating mode changes from shutdown or Rx to TxRx or Tx, or when a new neighbor is detected (namely new LLDPDUs are received and no such neighbor information is stored locally), to allow the neighbor device to quickly study the information about this device, the fast sending mechanism will be initiated. The fast sending mechanism adjusts the LLDPDU sending interval to 1 second and continuously transmits a certain number of LLDPDUs.

Mechanism for Receiving LLDPDUs

A LLDP-enabled port operating in TxRx mode or Rx Only mode will be able to receive LLDPDUs, and will check the validity of received LLDPDUs to verify they are new neighbor information or updates of existing neighbor information. The neighbor information will be stored on the local device. Meanwhile, an aging timer will be set according to the value in TTL TLV carried in the LLDPDU. If the TTL value is zero, the information is aged out immediately.

Protocol Specifications

The protocols and standards related to LLDP include:

- IEEE 802.1AB 2005: Station and Media Access Control Connectivity Discovery

- ANSI/TIA-1057: Link Layer Discovery Protocol for Media Endpoint Devices

Configuring LLDP Basic Functions

Function	**Default setting**
Globally enable LLDP	Enabled
Enable LLDP on the port	Enabled
Operating mode of LLDP	TxRx
Port re-initialization delay	2 seconds
LLDPDU transmit interval	30 seconds
LLDPDU transmit delay	2 seconds
Neighbor information aging timer	120 seconds
LLDPDU encapsulation format	Ethernet II
Enable LLDP Trap	Disabled
LLDP error detection	Enabled

Enabling LLDP

By default, LLDP is enabled globally and on each port. To make LLDP take effect on certain ports, you must enable LLDP both globally and on these ports.

Execute the following steps to disable LLDP globally and on each port.

Command	**Function**
Ruijie(config)#**no lldp enable**	Disable LLDP globally.

Ruijie(config)# **interface** *interface-name*	Enter interface configuration mode. LLDP runs on the actual physical interface (for AP port, it runs on AP's member port). LLDP is not supported on stacking port or VSL port.
Ruijie(config-if)#**no lldp enable**	Disable LLDP on the interface.
Ruijie(config-if)#**show lldp status**	Display LLDP state.

To enable LLDP globally or on the port, execute "lldp enable" command.

Caution

Disabling the LLDP globally will disable LLDP on the device. Meanwhile, the device will send Shutdown LLDPDUs to neighbor devices in order to delete the corresponding LLDP information.

Note

The port can learn up to 5 neighbors.

If a neighbor device does not support the LLDP, but its downlink device does, the information of non-directly connected devices may be learnt on the port as the neighbor device may forward the LLDP packets.

Configuration example:

# Globally disable LLDP and display LLDP state.

```
Ruijie(config)#no lldp enable

Ruijie(config)#show lldp status

Global status of LLDP: Disable
```

Configuring LLDP Operating Mode

By default, LLDP is enabled on the interface and operates in TxRx mode. The user can change the operating mode to Tx mode or Rx mode as needed. Execute the following steps to configure LLDP operating mode.

Command	**Function**
Ruijie(config)# **interface** *interface-name*	Enter interface configuration mode. LLDP runs on the actual physical interface (for AP port, it runs on AP's member port). LLDP is not supported on stacking port or VSL port.

Ruijie(config-if)#**lldp mode** { **tx** | **rx** | **txrx** }	Configure LLDP operating mode. The configurable operating modes include Tx, Rx and TxRx.
Ruijie(config-if)#**show lldp status interface** *interface-name*	Display LLDP state on the interface.

Configuration example:

# Configure LLDP operating mode as Tx on the interface and display LLDP state on the interface

```
Ruijie(config)#interface gigabitethernet 0/1
Ruijie(config-if)#lldp mode tx
Ruijie(config-if)#show lldp status interface gigabitethernet 0/1
Port [GigabitEthernet 0/1]
Port status of LLDP           : Enable
Port state                   : UP
Port encapsulation            : Ethernet II
Operational mode              : TxOnly
Notification enable           : NO
Error detect enable           : YES
Number of neighbors           : 0
Number of MED neighbors        : 0
```

Configuring the Advertisable TLVs

By default, all TLVs other than Location Identification TLV can be advertised on the interface. Execute the following steps to configure advertisable TLVs on the interface.

Command	**Function**

Ruijie(config)# **interface** *interface-name*	Enter interface configuration mode. LLDP runs on the actual physical interface (for AP port, it runs on AP's member port). LLDP is not supported on stacking port or VSL port.
Ruijie(config-if)# **lldp tlv-enable** { **basic-tlv** { **all** | **port-description** | **system-capability** | **system-description** | **system-name** } | **dot1-tlv** { **all** | **port-vlan-id** | **protocol-vlan-id** [*vlan-id*] | **vlan-name** [*vlan-id*] } | **dot3-tlv** { **all** | **link-aggregation** | **mac-physic** | **max-frame-size** | **power** } | **med-tlv** { **all** | **capability** | **inventory** | **location** { **civic-location** | **elin** } **identifier** *id* | **network-policy profile** [*profile-num*] | **power-over-ethernet** } }	Configure the TLV types that the interface allows the port to advertise. By default, all TLVs other than Location Identification TLV can be advertised on the interface.
Ruijie(config-if)# **show lldp tlv-config interface** *interface-name*	Display the attributes of advertisable TLVs.

Note

When configuring basic management TLVs, IEEE 802.1 organizationally specific TLVs and IEEE 802.3 organizationally specific TLVs, if "all" parameter is specified, all corresponding optional TLVs will be advertised.

When configuring LLDP-MED TLVs, if "all" parameter is specified, all LLDP-MED TLVs other than Location Identification TLV will be advertised.

Configure to allow the advertisement of LLDP-MED MAC/PHY TLVs before that of LLDP-MED Capability TLVs.

Configure to cancel the advertisement of LLDP-MED Capability TLVs before that of LLDP-MED MAC/PHY TLVs.

When configuring LLDP-MED TLVs, the LLDP-MED Capability TLV shall be configured as advertisable in order to further configure other LLDP-MED TLVs as advertisable.

In order not to advertise LLDP-MED Capability TLV, other LLDP-MED TLVs shall be configured as non-advertisable, so that LLDP-MED TLVs are not advertised.

For the meaning of respective key words of "lldp tlv-enable", please refer to the descriptions given in "LLDP-CREF".

When associating the device with an IP phone, you can configure the network policy TLV delivery policy to the IP phone if it supports LLDP-MED. Then, the IP phone modifies the voice flow tag and QoS. At this time, the voice VLAN function is not required, but it is required to configure the port connecting to the IP phone as the QoS trusted port. If the IP phone does not support LLDP-MED, the voice VLAN configuration is required and the phone MAC address must be manually configured to the voice VLAN OUI list.

Configuration example:

# Configure to disable the advertisement of Port And Protocol VLAN ID TLV specified by IEEE 802.1.

```
Ruijie(config)#interface gigabitethernet 0/1

Ruijie(config-if)#no lldp tlv-enable dot1-tlv protocol-vlan-id

Ruijie(config-if)#show lldp tlv-config interface gigabitethernet 0/1

LLDP tlv-config of port [GigabitEthernet 0/1]

            NAME               STATUS DEFAULT

------------------------------ ------ -------

Basic optional TLV:

Port Description TLV            YES   YES

System Name TLV                 YES   YES

System Description TLV          YES   YES

System Capabilities TLV         YES   YES

Management Address TLV          YES   YES



IEEE 802.1 extend TLV:

Port VLAN ID TLV                YES   YES

Port And Protocol VLAN ID TLV   NO    YES

VLAN Name TLV                   YES   YES



IEEE 802.3 extend TLV:

MAC-Physic TLV                  YES   YES
```

```
Power via MDI TLV               YES    YES

Link Aggregation TLV             YES    YES

Maximum Frame Size TLV           YES    YES



LLDP-MED extend TLV:

Capabilities TLV                 YES    YES

Network Policy TLV               YES    YES

Location Identification TLV      NO     NO

Extended Power via MDI TLV       YES    YES

Inventory TLV                    YES    YES
```

Configuring the Management address Advertised in LLDPDU

The management address of a device is used by the network management system to identify and manage the device.

Execute the following steps to configure the management address to be advertised in LLDPDU:

Command	Function
Ruijie(config)# **interface** *interface-name*	Enter interface configuration mode. LLDP runs on the actual physical interface (for AP port, it runs on AP's member port). LLDP is not supported on stacking port or VSL port.
Ruijie(config-if)#**lldp management-address-tlv** [*ip-address*]	Configure the management address advertised in the LLDP packet.
Ruijie(config-if)#**show lldp local-information interface** *interface-name*	Display LLDP local information about a specific interface.

Note

By default, the management address is advertised in LLDPDU, and is the IPv4 address of the lowest-ID VLAN carried on the port. If IPv4 address is not configured for this VLAN, the next lowest-ID VLAN carried on the port will be tried until the IPv4 address is obtained.

If the IPv4 address is still not found, the IPv6 address of the lowest-ID VLAN carried on the port will be tried.

If the IPv6 address is still not found, the MAC address of the device will be advertised as the management address.

Configuration example:

# Configure the management address advertised in LLDPDU as 192.168.1.1 and display the corresponding configuration.

```
Ruijie(config)#interface gigabitethernet 0/1
Ruijie(config-if)#lldp management-address-tlv 192.168.1.1
Ruijie(config-if)#show  lldp local-information interface GigabitEthernet 0/1
Lldp local-information of port [GigabitEthernet 0/1]
   Port ID type                   : Interface name
  Port id                       : GigabitEthernet 0/1
  Port description                :

  Management address subtype       : ipv4
  Management address              : 192.168.1.1
  Interface numbering subtype      : ifIndex
  Interface number                : 0
  Object identifier               :

  802.1 organizationally information
  Port VLAN ID                   : 1
  Port and protocol VLAN ID(PPVID)  : 1
     PPVID Supported             : YES
     PPVID Enabled               : NO
```

```
 VLAN name of VLAN 1             : VLAN0001

 Protocol Identity              :



 802.3 organizationally information

 Auto-negotiation supported       : YES

 Auto-negotiation enabled         : YES

 PMD auto-negotiation advertised   : 1000BASE-T full duplex mode, 100BASE-TX full duplex mode,
100BASE-TX half duplex mode, 10BASE-T full duplex mode, 10BASE-T half duplex mode

 Operational MAU type            : dot3MauType100BaseTXFD: 2 pair category 5 UTP, full duplex
mode

 PoE support                    : NO

Link aggregation supported       : YES

 Link aggregation enabled         : NO

 Aggregation port ID             : 0

 Maximum frame Size              : 1500



 LLDP-MED organizationally information

 Power-via-MDI device type        : PD

 Power-via-MDI power source       : Local

 Power-via-MDI power priority     :

 Power-via-MDI power value        :

 Model name                     : Model name
```

Configuring the Number of Fast Sent LLDPDUs

When a new neighbor is detected or when LLDP operating mode changes from shutdown or Rx to TxRx or Tx, to allow the neighbor device to quickly study the information about this device, the fast sending mechanism will be initiated. The fast sending mechanism shortens the LLDPDU sending interval to 1 second and continuously transmits a certain number of LLDPDUs before restoring to the normal transmit interval.

Command	**Function**
Ruijie(config)#**lldp fast-count** *count*	Configure the number of fast sent LLDPDUs in the range from 1 to 10. The default is 3.
Ruijie(config-if)#**show lldp status**	Display LLDP state.

Configuration example:

# Configure the number of fast sent LLDPDUs to 5.

```
Ruijie(config)#lldp fast-count 5
Ruijie(config)#show lldp status
Global status of LLDP               : Enable
Neighbor information last changed time :
Transmit interval                   : 30s
Hold multiplier                     : 4
Reinit delay                       : 2s
Transmit delay                     : 2s
Notification interval               : 5s
Fast start counts                   : 5
```

Configuring TTL Multiplier and LLDPDU Transmit interval

The value of Time To Live TLV in LLDPDU = TTL multiplier × LLDPDU transmit interval + 1. Therefore, the TTL of local device information on the neighbor device can be controlled by adjusting TTL multiplier.

The LLDPDU transmit interval can be adjusted. Execute the following steps to configure TTL multiplier and LLDPDU transmit interval.

Command	**Function**

Ruijie(config)#**lldp hold-multiplier** *value*	Configure TTL multiplier in the range from 2 to 10. The default is 4.
Ruijie(config)#**lldp timer tx-interval** *seconds*	Configure LLDPDU transmit interval in the range from 5 to 32768 in the unit of seconds. The default is 30.
Ruijie(config-if)#**show lldp status**	Display LLDP state.

Configuration example:

# Configure TTL multiplier to 3 and LLDPDU transmit interval to 20 seconds. By this time, the TTL of local device information on the neighbor device is 61 seconds.

```
Ruijie(config)#lldp hold-multiplier 3

Ruijie(config)#lldp timer tx-interval 20

Ruijie(config)#show lldp status

Global status of LLDP               : Enable

Neighbor information last changed time :

Transmit interval                   : 20s

Hold multiplier                    : 3

Reinit delay                       : 2s

Transmit delay                     : 2s

Notification interval               : 5s

Fast start counts                   : 3
```

Configuring LLDPDU Transmit Delay

An LLDP-enabled port will send LLDPDUs when the local device information changes. To avoid frequent LLDPDU sending during times of frequent local device information change, we can configure LLDPDU transmit delay to control the frequent transmission of LLDPDUs. The default transmit delay is 2 seconds. Execute the following steps to configure the LLDPDU transmit delay.

Command	**Function**

Ruijie(config)#**lldp timer tx-delay** *seconds*	Configure LLDPDU transmit delay
Ruijie(config)#**show lldp status**	Display LLDP state.

Configuration example:

# Configure LLDPDU transmit delay to 3 seconds and display LLDP state.

```
Ruijie(config)#lldp timer tx-delay 3

Ruijie(config)#show lldp status

Global status of LLDP               : Enable

Neighbor information last changed time :

Transmit interval                   : 30s

Hold multiplier                    : 4

Reinit delay                       : 2s

Transmit delay                     : 3s

Notification interval               : 5s

Fast start counts                   : 3
```

Configuring Port Re-initialization Delay

When the LLDP operating mode of a port changes, the port will initialize the protocol state machine. To prevent LLDP from being initialized too frequently during times of frequent operating mode change, you can configure port re-initialization delay. Execute the following steps to configure port re-initialization delay:

Command	**Function**
Ruijie(config)#**lldp timer reinit-delay** *seconds*	Configure port re-initialization delay.
Ruijie(config)#**show lldp status**	Display LLDP state.

Configuration example:

# Configure the port re-initialization delay to 3 seconds and display LLDP state.

```
Ruijie(config)#lldp timer reinit-delay 3

Ruijie(config)#show lldp status

Global status of LLDP               : Enable

Neighbor information last changed time :

Transmit interval                   : 30s

Hold multiplier                    : 4

Reinit delay                       : 3s

Transmit delay                     : 2s

Notification interval               : 5s

Fast start counts                   : 3
```

Configuring LLDP Trap

By configuring LLDP Trap, the LLDP information of local device (such as information about the detection of new neighbor or the fault on the communication link) can be sent to the network management server. The administrator can monitor the network operation status according to such information.

To prevent excessive LLDP traps from being sent, you can set an interval for sending LLDP Traps. If LLDP information change is detected during this interval, traps will be sent to the network management server.

By default, LLDP Trap is disabled.

Execute the following steps to configure LLDP Trap:

Command	Function
Ruijie(config)#**lldp timer notification-interval** *seconds*	Configure the interval for sending LLDP Traps in the range from 5 to 3600 in the unit of seconds. The default is 5.
Ruijie(config)# **interface** *interface-name*	Enter interface configuration mode. LLDP runs on the actual physical interface (for AP port, it runs on AP's member port). LLDP is not supported on stacking port or VSL port.
Ruijie(config-if)#**lldp notification remote-change enable**	Enable LLDP Trap. LLDP Trap is disabled by default.

Ruijie(config-if)#**show lldp status**	Display LLDP state.

Configuration example:

# Enable LLDP Trap and configure the interval for sending LLDP Traps to 10 seconds.

```
Ruijie(config)#lldp timer notification-interval 10
Ruijie(config)#interface gigabitethernet 0/1
Ruijie(config-if)#lldp notification remote-change enable
Ruijie(config-if)#show lldp status
Global status of LLDP               : Enable
Neighbor information last changed time :
Transmit interval                   : 30s
Hold multiplier                    : 4
Reinit delay                       : 2s
Transmit delay                     : 2s
Notification interval               : 10s
Fast start counts                   : 3
------------------------------------------------------------
Port [GigabitEthernet 0/1]
------------------------------------------------------------
Port status of LLDP           : Enable
Port state                   : UP
Port encapsulation            : Ethernet II
Operational mode              : RxAndTx
Notification enable           : YES
```

```
Error detect enable           : YES
Number of neighbors           : 0
Number of MED neighbors        : 0
```

Configuring LLDP Error Detection

Configure LLDP error detection, including the detection of VLAN configurations on both sides of the link, port state detection, port aggregation configuration detection, MTU configuration detection and loop detection. If any error is detected by LLDP, LOG information will be printed to notify the administrator.

Execute the following steps to configure LLDP error detection:

Command	**Function**
Ruijie(config)# **interface** *interface-name*	Enter interface configuration mode. LLDP runs on the actual physical interface (for AP port, it runs on AP's member port). LLDP is not supported on stacking port or VSL port.
Ruijie(config-if)#**lldp error-detect**	Configure LLDP error detection. LLDP error detection is enabled by default.
Ruijie(config-if)#**show lldp status interface** *interface-name*	Display LLDP state on the interface.

Configuration example:

# Configure LLDP error detection.

```
Ruijie(config)#interface gigabitethernet 0/1
Ruijie(config-if)#lldp error-detect
Ruijie(config-if)#show lldp status interface gigabitethernet 0/1
Port [GigabitEthernet 0/1]
Port status of LLDP           : Enable
Port state                   : UP
Port encapsulation            : Ethernet II
Operational mode              : RxAndTx
```

```
Notification enable           : NO

Error detect enable           : YES

Number of neighbors           : 0

Number of MED neighbors        : 0
```

Configuring LLDPDU Encapsulation Format

By default, LLDPDUs are encapsulated in Ethernet II frames. The configurable encapsulation formats include Ethernet II and SNAP.

When configured to Ethernet II format, the device can only send and receive Ethernet II-encapsulated LLDP packets.

When configured to SNAP format, the device can only send and receive SNAP-encapsulated LLDP packets.

Execute the following steps to configure LLDPDU encapsulation format:

Command	**Function**
Ruijie(config)# **interface** *interface-name*	Enter interface configuration mode. LLDP runs on the actual physical interface (for AP port, it runs on AP's member port). LLDP is not supported on stacking port or VSL port.
Ruijie(config-if)#**lldp encapsulation snap**	Configure LLDPDU encapsulation format to SNAP.
Ruijie(config-if)#**show lldp status interface** *interface-name*	Display LLDP state on the interface.

Caution

To guarantee normal communication between local device and neighbor device, the same LLDPDU encapsulation format must be used.

Configuration example:

# Configure LLDPDU encapsulation format to SNAP and display the corresponding configuration.

```
Ruijie(config)#interface gigabitethernet 0/1

Ruijie(config-if)#lldp encapsulation snap

Ruijie(config-if)#show lldp status interface gigabitethernet 0/1
```

```
Port [GigabitEthernet 0/1]
Port status of LLDP           : Enable
Port state                   : UP
Port encapsulation            : Snap
Operational mode              : RxAndTx
Notification enable           : NO
Error detect enable           : YES
Number of neighbors           : 0
Number of MED neighbors        : 0
```

Displaying and Clearing Configurations

Command	Function
Ruijie(config)# **lldp network-policy profile** *profile-num*	Enter the LLDP network-policy configuration mode.
Ruijie(config-lldp-network-policy)# { **voice** | **voice-signaling** } **vlan** { { *vlan-id* [**cos** ***cvalue*** | **dscp** *dvalue*] } | { **dot1p** [**cos** ***cvalue*** | **dscp** *dvalue*] } | **none** | **untagged** } **no** { **voice** | **voice-signaling** } **vlan**	Configure the LLDP network-policy.

Configuration example:

# Configure the LLDP packet advertised from interface 1 as follows:
Network Policy TLV: **1**
voice VLAN ID: **3**
cos: **4**
dscp: **6**

```
Ruijie#config
Ruijie(config)#lldp network-policy profile 1
Ruijie(config-lldp-network-policy)# voice vlan 3 cos 4
Ruijie(config-lldp-network-policy)# voice vlan 3 dscp 6

Ruijie(config-lldp-network-policy)#exit

Ruijie(config)# interface gigabitethernet 0/1

Ruijie(config-if-GigabitEthernet 0/1)# lldp tlv-enable med-tlv network-policy profile 1
```

Configuring the Civic Address Information of a Device

Run the commands listed in the following table to configure the address information of a device.

Command	**Function**
Ruijie(config)# **lldp location civic-location identifier** *id*	Enters the LLDP Civic Address configuration mode
Ruijie(config-lldp-civic**)# device-type** *device-type*	Configure the device type. The default device type is a switch.
Ruijie(config-lldp-civic)# **civic-location** { **country** | **state** | **county** | **city** | **division** | **neighborhood** | **street-group** | **leading-street-dir** | **trailing-street-suffix | street-suffix** | **number** | **street-number-suffix** | **landmark** | **additional-location-information** | **name** | **postal-code** | **building** | **unit** | **floor** | **room** | **type-of-place |** **postal-community-name** | **post-office-box** | **additional-code** } *ca-word*	Configure the LLDP civic address information.

Configuration example:

# Configure the address of device interface 1 as follows:
Device type: switch
Country: CH
City: Fuzhou
Postal-code: 350000

```
Ruijie#config

Ruijie(config)#lldp location civic-location identifier 1

Ruijie(config-lldp-civic)# country CH

Ruijie(config-lldp-civic)# city Fuzhou
Ruijie(config-lldp-civic)# postal-code 350000

Ruijie(config-lldp-civic)# exit

Ruijie(config)# interface gigabitethernet 0/1

Ruijie(config-if-GigabitEthernet 0/1)# lldp tlv-enable location civic-location identifier 1
```

Configuring the Emergency Call Number

Run the commands listed in the following table to configure the emergency call number of a device.

Command	Function
Ruijie(config)# **lldp location elin identifier** *id* **elin-location** *tel-number*	Configure the emergency call number.

Configuration example:

# Configure the emergency call number of device interface 1 as 085285555556.

```
Ruijie#config

Ruijie(config)#lldp location elin identifier 1 elin-location 085283671111

Ruijie(config)# interface gigabitethernet 0/1

Ruijie(config-if-GigabitEthernet 0/1)# lldp tlv-enable location elin identifier 1
```

Viewing and Clearing Configurations

Command	Function
show lldp local-information [**global** | **interface** *interface-name*]	Show the device information to be sent to a neighbor.
show lldp location { **civic-location** | **elin** } { **identifier** ***id*** | **interface** *interface-name* | **static** }	Show the civic address information or emergency call number of a local device.
show lldp neighbors [**interface** *interface-name*] [**detail**]	Show the device information about an adjacent neighbor.

show lldp network-policy profile [*profile-num*]	Show the LLDP network-policy configuration.
show lldp statistics [**global** | **interface** *interface-name*]	Show the LLDP statistics.
show lldp status [**interface** *interface-name*]	Show the LLDP status.
show lldp tlv-config [**interface** *interface-name*]	Show the optional TLVs that can be advertised.
clear lldp statistics [**interface** *interface-name*]	Clear LLDP statistics.
clear lldp table [**interface** *interface-name*]	Clear the information about LLDP neighbors.

Configuration example:

# Show the device information about an adjacent neighbor connecting a specified port.

```
Ruijie# show lldp neighbors detail

Lldp neighbor-information of port [GigabitEthernet 0/1]

  Neighbor index                 : 1

  Device type                   : LLDP Device

  Update time                   : 12minutes 40seconds

Aging time                      : 5seconds

  Chassis ID type                : MAC address

  Chassis id                    : 00d0.f822.33cd

  System name                   : System name

  System description            : System description
  System capabilities supported    : Repeater, Bridge, Router
  System capabilities enabled      : Repeater, Bridge, Router

  Management address subtype       : 802 mac address
  Management address              : 00d0.f822.33cd
  Interface numbering subtype      :
  Interface number               : 0
  Object identifier              :

  LLDP-MED capabilities           :
  Device class                  :
  HardwareRev                   :
  FirmwareRev                   :
  SoftwareRev                   :
  SerialNum                     :
  Manufacturer name              :
  Asset tracking identifier        :

  Port ID type                  : Interface name
  Port id                      : GigabitEthernet 0/2
  Port description               :

  802.1 organizationally information
```

```
  Port VLAN ID                    : 1
  Port and protocol VLAN ID(PPVID)  : 1
      PPVID Supported             : YES
      PPVID Enabled               : NO
  VLAN name of VLAN 1              : VLAN0001
  Protocol Identity               :
  802.3 organizationally information
  Auto-negotiation supported        : YES
  Auto-negotiation enabled         : YES
  PMD auto-negotiation advertised   : 1000BASE-T full duplex mode, 100BASE-TX full duplex mode,
100BASE-TX half duplex mode, 10BASE-T full duplex mode, 10BASE-T half duplex mode
  Operational MAU type             : speed(100)/duplex(Full)
  PoE support                     : NO
  Link aggregation supported        : YES
  Link aggregation enabled         : NO
  Aggregation port ID              : 0
  Maximum frame Size               : 1500

  LLDP-MED organizationally information
  Power-via-MDI device type         :
  Power-via-MDI power source        :
  Power-via-MDI power priority      :
  Power-via-MDI power value         :
```

Note For details about LLDP output information, see the description in *LLDP Command Reference*.

Typical LLDP Configuration Examples

Use LLDP to View Topological Connections

Networking Requirements

Devices required

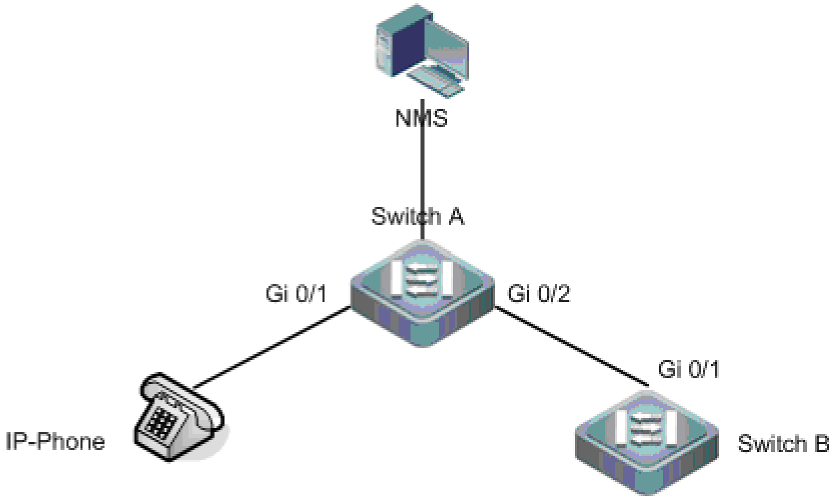
Two Ethernet switches (Switch A and Switch B), one MED device (taking IP Phone as the example) and one NMS (Network management System).

Configuration required

LLDP is enabled by default. No further configuration is needed.

Network Tpology

Fig 4 Basic topological diagram of LLDP



Configuration Tips

- LLDP operating mode on the port is TxRx.

- LLDPDU transmit times will use default values, namely LLDPDU transmit interval is 30 seconds and LLDPDU transmit delay is 2 seconds.

Configuration Steps

By default, LLDP is enabled, and no further configuration is needed.

Verification

- Display the information about the neighbor device connecting with Switch A.

# Display the information about the neighbor device on Switch A.

```
Ruijie# show lldp neighbors gigabitethernet 0/2

Capability codes:

    (R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device

    (W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other



Local Intf  Port ID  Capability   Aging-time

Gi 0/2      Gi 0/1   B,R          120



Total entries displayed: 1
```

The above messages show that the MAC address of neighbor device connected to port 2 of switch A is 00d0-f822-33cd and the port connected is Gi 0/1. The neighbor device allows bridging and routing.

# Display the detailed information about the neighbor device connected to port Gi 0/2 of Switch A.

```
Ruijie# show lldp neighbor-information interface gigabitethernet 0/2

Lldp neighbor-information of port [GigabitEthernet 0/2]

  Neighbor index                  : 1

  Device type                    : LLDP Device

  Update time                    : 5minute 39second



  Chassis ID type                 : MAC address

  Chassis id                     : 00d0.f822.33cd

  System name                    : System name
  System description              : System description
  System capabilities supported    : Repeater, Bridge, Router
  System capabilities enabled     : Repeater, Bridge, Router

  Management address subtype      : 802 mac address
  Management address              : 00d0.f822.33cd
  Interface numbering subtype     :
  Interface number               : 0
  Object identifier               :


  LLDP-MED capabilities           :
  Device class                   :
  HardwareRev                    :
  FirmwareRev                    :
  SoftwareRev                    :
  SerialNum                     :
  Manufacturer name               :
  Asset tracking identifier        :

  Port ID type                   : Interface name
  Port id                       : GigabitEthernet 0/1
  Port description                :
  802.1 organizationally information
  Port VLAN ID                    : 1
  Port and protocol VLAN ID(PPVID)  : 1
     PPVID Supported             : YES
     PPVID Enabled               : NO
  VLAN name of VLAN 1             : VLAN0001
  Protocol Identity               :



  802.3 organizationally information
  Auto-negotiation supported       : YES
  Auto-negotiation enabled         : YES
  PMD auto-negotiation advertised   : 1000BASE-T full duplex mode, 100BASE-TX full duplex mode,
 100BASE-TX half duplex mode, 10BASE-T full duplex mode, 10BASE-T half duplex mode
  Operational MAU type            : dot3MauType1000BaseTFD: Four-pair Category 5 UTP, full
 duplex mode
  PoE support                    : NO
  Link aggregation supported       : YES
  Link aggregation enabled         : NO
  Aggregation port ID             : 0
  Maximum frame Size              : 1500


  LLDP-MED organizationally information
  Power-via-MDI device type       :

  Power-via-MDI power source      :

  Power-via-MDI power priority     :

            Power-via-MDI power value        :
```

Use LLDP Error Detection Feature to Perform Error Detection

Networking Requirements

- Devices required

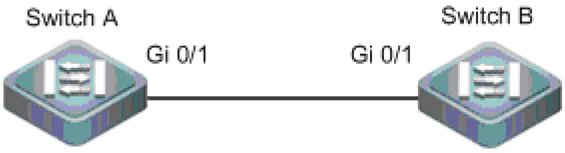
two Ethernet switches (Switch A and Switch B)

- Configuration required

LLDP is enabled by default. No further configuration is needed.

Network Topology

Fig 5 Basic topological diagram of LLDP



Configuration Tips

- LLDP operating mode on the port is TxRx.
- LLDPDU transmit times will use default values, namely LLDPDU transmit interval is 30 seconds and LLDPDU transmit delay is 2 seconds.
- LLDP error detection is enabled by default. No further configuration is needed.

Configuration Steps

1. Configure the bit-rate of port Gi 0/1 of Switch A to 100M.

```
Ruijie#config

Ruijie(config)#interface gigabitethernet 0/1

Ruijie(config-if)#speed 100

%Warning: the speed/duplex of port GigabitEthernet 0/1 may not match with it's neighbor.
```

The above messages show that bit-rate and duplex capabilities of port 1 may not match with that of port on neighbor device.

Verification

While the administrator is carrying out VLAN configuration, port bit-rate and duplex configuration, aggregation port configuration and port MTU configuration, if the information doesn't match with the configurations of neighbor device the corresponding error messages will be prompted.

Configuring VRRP

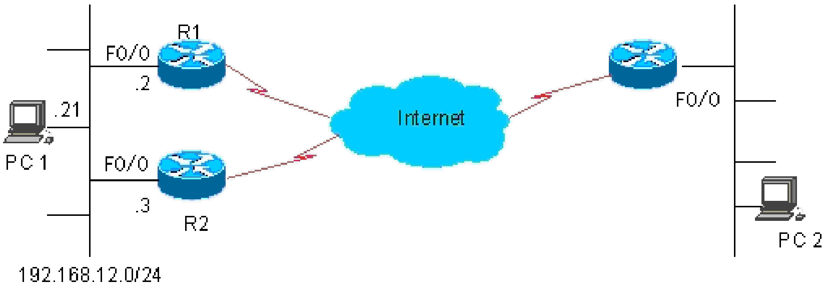
Overview

The Virtual Router Redundancy Protocol (VRRP) is designed to work in master/backup mode, so that traffic can switch over to a backup router without affecting internal or external data communication when the master router fails. In this process, parameters of the internal network do not need to be modified. Multiple routers in a VRRP group map to one virtual router. VRRP ensures that only one router transmits packets at a time, whereas hosts send data packets to the virtual router. The router that forwards data packets is elected as the master router. If the master router cannot work due to certain reasons at a time, a backup router is used to perform tasks of the original master router. Using VRRP allows all hosts in a local area network (LAN) looking like using only one router, and guarantees route connectivity even if the first-hop router fails.

RFC 2338, RFC 3768, and RFC 5798 define the format and operating mechanism of VRRP packets. A VRRP packet is a multicast packet with a specified destination address. It is sent by the master router to indicate that the master router is running properly or used to elect the master router. VRRP allows another router to automatically take over the router that fails to support the routing and forwarding function in an IP LAN, therefore implementing the hot backup and error tolerance of IP routes. VRRP also guarantees the communication continuity and reliability of hosts inside the LAN. One VRRP group consists of multiple routers in redundancy mode. At any moment, however, only one router serves as the master router to perform routing and forwarding functions. All other routers in the VRRP group are backup routers. The switchover between routers in the VRRP group is completely transparent to hosts in the LAN. RFC defines the following router switchover rules:

- VRRP elects the master router using a simple election method. First, it compares the VRRP priority of interfaces on various routers in the VRRP group and elects the router with the highest interface priority as the master router. The status of the elected router changes to Master. If the routers have the same priority, VRRP compares the primary IP address of network interfaces of the routers. The router with the biggest primary IP address becomes the master router to provide actual routing and forwarding services.
- After the master router is elected, other routers serve as backup routers whose status changes to Backup. These backup routers monitor the status of the master router by receiving VRRP packets that are periodically sent by the master router. When working normally, the master router sends a VRRP multicast packet that is known as an advertisement packet at a certain interval to inform the backup routers that the master router itself is running properly. If a backup router in the VRRP group does not receive any advertisement packet from the master router within the specified time, it sets its own status to Master. If multiple routers in the VRRP group exist in Master state, an election process as described previously is implemented again so that a router with the highest priority is selected as the new master router to implement the backup function of the VRRP.

Figure 1 Operating principles of VRRP



Once the master router is selected out of the VRRP group, the packets of all hosts in the LAN are routed and forwarded by the master router. Figure 1 shows the specific communication process. Routers R1 and R2 are connected to the LAN segment 192.168.12.0/24 through the Ethernet interface F0/0. VRRP is enabled on the Ethernet interface F0/0 of routers R1 and R2. The virtual router IP address of the VRRP group is set as the default gateway on all hosts in the LAN. Hosts in the LAN can detect only the virtual router of the VRRP group, whereas the master router that practically performs routing and forwarding functions is transparent to all of them. For example, the host PC 1 in the LAN sends a data packet to PC 2 by using the virtual router of the VRRP group as the default gateway so as to communicate with PC 2 in the same LAN. Upon receipt of the data packet, the master router of the VRRP group forwards the data packet to PC 2. In this communication process, PC 1 can detect only the virtual router but does not know whether R1 or R2 is the master router that plays the role of the virtual router. The master router of the VRRP group is selected between R1 and R2. Once the master router fails, the other router takes over traffic and becomes the new master router.

RFC 5798 redefines the format of a VRRP packet. The RGOS IPv6 VRRP complies with RFC 5798. In later descriptions, RGOS IPv6 VRRP is called VRRPv3 for short whereas the original VRRP implementation is simply called VRRPv2. In IPv4 VRRP, VRRPv2 and VRRPv3 are strictly distinguished from each other. The two VRRP standards define different fields in a VRRP packet. For this reason, RGOS IPv4 VRRP supports VRRPv3 and provide compatibility with VRRPv2. In contrast, IPv6 does not distinguish VRRPv2 from VRRPv3, because IPv6 VRRP is defined only in VRRPv3.

Currently, VRRP is defined in the following three protocols:

- RFC 2338
- RFC 3768
- RFC 5798

RFC 3768 is an update of RFC 2338 and defines mechanisms such as IPv4 VRRP. RFC 5798 is an improvement and extension of RFC 3768, and defines IPv4 VRRP and IPv6 VRRP.

Caution To provide compatibility with widely deployed devices that do not support VRRPv3, IPv6 VRRP, however, invariably uses VRRPv3.

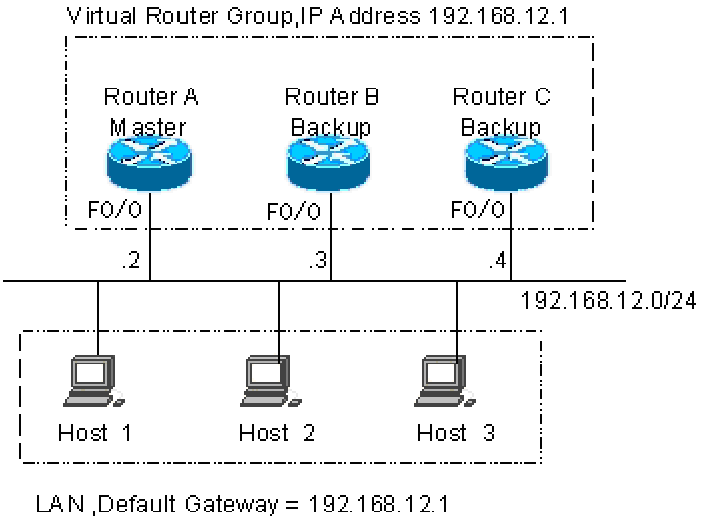
VRRP Application

VRRP supports two application modes: basic applications and advanced applications. In basic applications, only one VRRP group is used to implement simple route redundancy. In advanced applications, multiple VRRP groups are used to implement route redundancy and load balancing.

Route Redundancy

Figure 2 shows an example of basic VRRP applications.

Figure 2 Example of basic VRRP applications

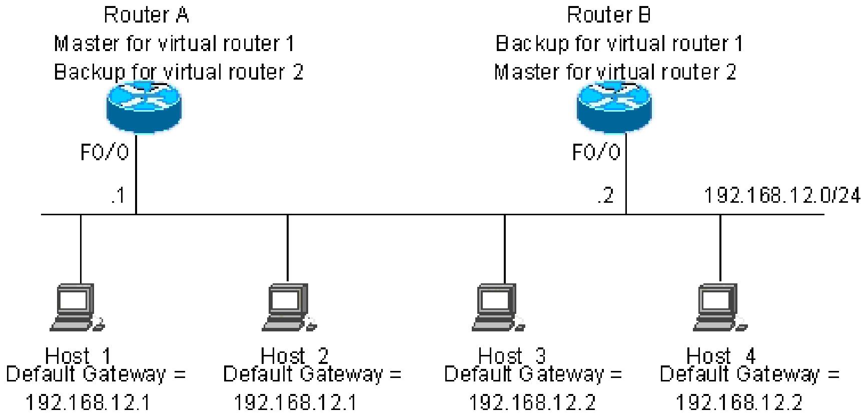


As shown in Figure 2, routers A, B, and C are connected to a LAN through Ethernet interfaces on which VRRP is enabled. Routers A, B, and C belong to the same VRRP group. The virtual IP address of the VRRP group is 192.168.12.1. Router A is elected as the master router of the VRRP group, whereas routers B and C are backup routers. The virtual router IP address 192.168.12.1 is set as the default gateway on hosts 1, 2, and 3 in the LAN. Data packets from hosts in the LAN to other networks are routed and forwarded by the master router A. If router A fails, a new master router is elected between routers B and C to route and forwards packets as a virtual router, therefore implementing simple route redundancy.

Load Balancing

Figure 3 shows an example of advanced VRRP applications.

Figure 3 Example of advanced VRRP applications



As shown in Figure 3, two virtual routers are set. For virtual router 1, the IP address 192.168.12.1 of the Ethernet interface F0/0 on router A is set as the IP address of the virtual router, so router A is the master router and router B is a backup router. For virtual router 2, the IP address 192.168.12.2 of the Ethernet interface F0/0 on router B is set as the IP address of the virtual router, so router B is the master router and router A is a backup router. The IP address 192.168.12.1 of virtual router 1 is set as the default gateway on hosts 1 and 2, and the IP address 192.168.12.2 of virtual router 2 is set as the default gateway on hosts 3 and 4 in the LAN. In this VRRP application example, routers A and B back up each other to implement route redundancy and share traffic from the LAN to implement load balancing.

Configuring VRRP

VRRP Configuration Task List

VRRP is applicable to multicast or broadcast LANs, such as Ethernets. VRRP configurations are mostly Ethernet interface configurations and involve the following configuration tasks:

- Enabling the VRRP function (Mandatory)
- Setting the authentication string of the VRRP group (Optional)
- Setting the advertisement interval of the VRRP group (Optional)
- Setting the preemption mode of the router in the VRRP group (Optional)
- Setting the Accept_Mode of the IPv6 VRRP virtual router
- Setting the priority of the router in the VRRP group (Optional)
- Setting the tracked interface of the VRRP group (Optional)
- Setting the tracked IP address of the VRRP group (Optional)
- Setting the periodic learning function of VRRP advertisement packets (Optional)
- Setting the description string of the VRRP group on the router (Optional)
- Setting the start delay of the VRRP group (Optional)
- Setting the IPv4 VRRP version (Optional)

You can determine which tasks to be configured based on your actual requirement.

Enabling the VRRP Function

You can add a VRRP group to a specific LAN segment by setting a group number and a virtual IPv4/IPv6 address for the VRRP group so as to enable the VRRP function on the corresponding Ethernet interface.

Command	Function
Ruijie(config-if)# **vrrp** *group* **ip** *ipaddress* [**secondary**]	Enables IPv4 VRRP.
Ruijie(config-if)# **no vrrp** *group* **ip** *ipaddress* [**secondary**]	Disables IPv4 VRRP.
Or:	
Ruijie(config-if)# **vrrp** *group* **ipv6** *ipv6-address*	Enables IPv6 VRRP.
Ruijie(config-if)# **no vrrp** *group* **ipv6** *ipv6-address*	Disables IPv6 VRRP.

The group number specified by the *group* parameter ranges from 1 to 255. If the virtual IP address is not specified, the router does not participate in the VRRP group. If the *secondary* parameter is not specified, the specified IP address becomes the primary IP address of the virtual router. The system does not identify whether an IPv6 address is a primary or secondary address. The first virtual IP address configured for an IPv6 VRRP group, however, must be a link-local address.

Caution

If the virtual IP address (primary or secondary) or virtual IPv6 address (link-local or non-link-local) of the VRRP group is consistent with the IP address (primary or secondary) or IPv6 address (link-local or non-link-local) of an Ethernet interface, the VRRP group is considered as owning the real IP address of the

Ethernet interface. In this case, the priority of the VRRP group is 255. If the Ethernet interface is available, the VRRP group is automatically in Master state.

☑ The NMX-2GEH line card supports listening to a maximum of 15 MAC addresses. The number of configurable VRRP groups depends on the number of MAC addresses supported by the current line card. If the number of configured VRRP groups is larger than the maximum number allowed by the line card, an error message is displayed. Note that the line card may also need to support MAC address listening according to other protocols, such as Open Shortest Path First (OSPF) and Routing Information Protocol (RIP).

Setting the Authentication String of the VRRP Group

VRRP supports two authentication modes: plain text authentication and no-authentication. When setting the authentication string of a VRRP group, you can also set the authentication mode of the VRRP group to plain text authentication or no-authentication. All members of the VRRP group must be set to the same authentication mode so as to normally communicate with one another. In plain text authentication mode, all routers in the VRRP group must have the same authentication password. The plain text authentication password does not guarantee security but is used only to prevent or prompt VRRP configuration errors.

Command	**Function**
Ruijie(config-if)# **vrrp group authentication** *string*	Sets the authentication string of the IPv4 VRRP group.
Ruijie(config-if)# **no vrrp group authentication**	Sets the authentication mode of the IPv4 VRRP group to no-authentication.

By default, the authentication mode of a VRRP group is no-authentication. If the plain text authentication mode is specified, the plain text authentication password consists of up to eight bytes.

⚠ Caution The authentication mode is already abandoned in RFC 5798 and no longer adopted in new specifications. Therefore, the user-specified authentication mode is applicable to VRRPv2 packets only.

Setting the Advertisement Interval of the VRRP Group

Command	**Function**
Ruijie(config-if)# **vrrp group timers advertise** { *advertise-interval* | **csec** *centisecond-interval* }	Sets the VRRP advertisement interval of the IPv4 VRRP master router.
Ruijie(config-if)# **no vrrp group timers advertise**	Restores the default VRRP advertisement interval of the IPv4 VRRP master router.
Or:	
Ruijie(config-if)# **vrrp ipv6 group timers advertise**{ *advertise-interval* | **csec** *centisecond-interval* }	Sets the VRRP advertisement interval of the IPv6 VRRP master router.
Ruijie(config-if)# **no vrrp ipv6 group timers advertise**	Restores the default VRRP advertisement interval of the IPv6 VRRP master router.

If the current router is the master router of the VRRP group, it sends VRRP advertisement packets at the set interval to advertise its own VRRP status, priority, and other information. By default, the master router sends VRRP advertisement packets at an interval of one second. The time for a VRRP backup router to switch over to the master router is defined in

RFC 2338, RFC 3768, and RFC 5798. It is three times the advertisement interval plus a Skew_Time. The Skew_Time is calculated with the following formula: Skew_Time = (((256 – Priority of the VRRP group) x VRRP advertisement interval) / 256). VRRPv3 supports a VRRP advertisement interval of the master router ranging from 50 to 99 milliseconds to accelerate VRRP convergence time without correlation with BFD. If the network traffic is heavy, an interval in milliseconds is not recommended, as the backup router may fail to receive the VRRP advertisement packets from the master router within the interval due to heavy traffic, causing status change.

Caution

If periodic VRRP learning is not enabled on routers, the same VRRP advertisement interval must be set on all routers in a VRRP group; otherwise, backup routers discard received VRRP advertisement packets.

Setting the Preemption Mode of the Router in the VRRP Group

A router in a VRRP group that works in preemption mode preempts other routers in the VRRP group to become the master router once detecting that its own priority is higher than the priority of the existing master router. If the VRRP group works in non-preemption mode, the router does not preempt other routers in the VRRP group to become the master router even when detecting that its own priority is higher than the priority of the existing master router. Setting the preemption mode is insignificant for a VRRP group whose virtual router address is an Ethernet interface IP address, because the router configured with the Ethernet interface IP address has the highest priority and automatically becomes the master router of the VRRP group.

Command	**Function**
Ruijie(config-if)# **vrrp** *group* **preempt** [**delay** *seconds]*	Sets the IPv4 VRRP group to the preemption mode.
Ruijie(config-if)# **no vrrp** *group* **preempt** [**delay]**	Sets the IPv4 VRRP group to the non-preemption mode or restores the default delay.
or	
Ruijie(config-if)# **vrrp ipv6** *group* **preempt** [**delay** *seconds]*	Sets the IPv6 VRRP group to the preemption mode.
Ruijie(config-if)# **no vrrp ipv6** *group* **preempt** [**delay]**	Sets the IPv6 VRRP group to the non-preemption mode or restores the default delay.

The optional parameter *delay seconds* defines a delay before a backup VRRP router advertises itself as the master router of the VRRP group. It is 0 seconds by default. Once the VRRP function is enabled, the VRRP group works in preemption mode by default.

Setting the Accept_Mode of the IPv6 VRRP Virtual Router

The Accept_Mode can be set for the IPv6 VRRP virtual router that serves as the master router to determine whether to receive and process packets destined to the IP address of the IPv6 VRRP virtual router itself. If the Accept_Mode is enabled, the IPv6 VRRP virtual router receives and processes packets destined to the IP address of the virtual router itself. If the Accept_Mode is not enabled, the IPv6 VRRP virtual router discards packets destined to the IP address of the virtual router itself but does not discard NA and NS packets. By default, the Accept_Mode is disabled. In addition, the IPv6 VRRP virtual router in Owner state receives and processes packets destined to the IP address of the virtual router itself, no matter whether the Accept_Mode is enabled or disabled.

Command	**Function**

Ruijie(config-if)# **vrrp ipv6** *group* **accept_mode**	Enables the Accept_Mode for an IPv6 VRRP group.
Ruijie(config-if)# **no vrrp ipv6** *group* **accept_mode**	Disables the Accept_Mode for an IPv6 VRRP group.

Setting the Priority of the Router in the VRRP Group

According to VRRP, the role of a router in a VRRP group is determined by the priority of the router in the VRRP group. A router in a VRRP group becomes the active or master router of the VRRP group if it works in preemption mode, has the highest priority, and has obtained a virtual IP address. The other routers that have a priority lower than the priority of the master router in the VRRP group becomes backup or listening routers. Once the VRRP function is enabled on a router, the priority of the router in a VRRP group is 100 by default.

Command	Function
Ruijie(config-if)# **vrrp** *group* **priority** *level*	Sets the priority of a router in an IPv4 VRRP group.
Ruijie(config-if)# **no vrrp** *group* **priority**	Restores the default priority of a router in an IPv4 VRRP group.
Or:	
Ruijie(config-if)# **vrrp ipv6** *group* **priority** *level*	Sets the priority of a router in an IPv6 VRRP group.
Ruijie(config-if)# **no vrrp ipv6** *group* **priority**	Restores the default priority of a router in an IPv6 VRRP group.

The priority defined by the *level* parameter ranges from 1 to 254. If the virtual IP address of a VRRP group is consistent with the real IP address of an Ethernet interface on the local router, the priority of the local router in the VRRP group is 255. In this case, the VRRP group configured on the router is automatically in Master state as long as the Ethernet interface is available, no matter whether the VRRP group works in preemption mode or not.

Setting the Tracked Interface of the VRRP Group

After a tracked interface is set for the VRRP group, the system dynamically adjusts the priority of the local router according to the status of the tracked interface. When the tracked interface becomes unavailable, the local router decreases its VRRP group priority based on settings. At this time, another router in the VRRP group may become the active or master router of the VRRP group if its status is more stable and its priority is higher.

Command	Function
Ruijie(config-if)# **vrrp** *group* **track** *interface-type number* [*interface-priority*]	Sets the tracked interface of an IPv4 VRRP group.
Ruijie(config-if)# **no vrrp** *group* **track** *interface-type number*	Cancels the tracked interface set for an IPv4 VRRP group.
Or:	
Ruijie(config-if)# **vrrp ipv6** *group* **track** *interface-type number* [*interface-priority*]	Sets the tracked interface of an IPv6 VRRP group.
Ruijie(config-if)# **no vrrp ipv6** *group* **track** *interface-type number*	Cancels the tracked interface set for an IPv6 VRRP group.

By default, no tracked interface is set for a VRRP group. The value of the *interface-priority* parameter ranges from 1 to 255. It is 10 by default if not specified.

Note The tracked interface can only be a reachable logical Layer 3 (L3) interface, such as a routed port, SVI, loopback interface, or tunnel interface.

Setting the Tracked IPv4/IPv6 Address of the VRRP Group

After a tracked IP address is set for the VRRP group, the system dynamically adjusts the priority of the local router depending on whether the tracked IP address is reachable. When the tracked IP address is unreachable and cannot be pinged, the local router decreases its VRRP group priority based on settings. At this time, another router in the VRRP group may become the active or master router of the VRRP group if its priority is higher. In the following commands, the optional parameter *interval* defines an interval at which the system detects whether the destination address is reachable, the optional parameter *timeout* defines a timeout interval that is used to determine that the destination is unreachable, and the optional parameter *retry* defines the number of retries when the destination is unreachable.

Command	Function
Ruijie(config-if)# **vrrp** *group* **track** *ip-address* [**interval** *interval-value]* [**timeout** *timeout-value]* [**retry** *retry-value]* [*priority]*	Sets the tracked IP address of an IPv4 VRRP group.
Ruijie(config-if)# **no vrrp** *group* **track** *ip-address*	Cancels the tracked IP address set for an IPv4 VRRP group.
Or:	
Ruijie(config-if)# **vrrp ipv6** *group* **track** { *ipv6-global-address* | { *ipv6-linklocal-address interface-type number* } } [**interval** *interval-value]* [**timeout** *timeout-value]* [**retry** *retry-value]* [*priority]*	Sets the tracked IP address of an IPv6 VRRP group.
Ruijie(config-if)# **no vrrp ipv6** *group* **track** { *ipv6-global-address* | { *ipv6-linklocal-address interface-type number* } }	Cancels the tracked IP address set for an IPv6 VRRP group.

By default, no tracked IP address is set for a VRRP group. The value of the *interval-value* parameter ranges from 1 to 3600 seconds. It is 3 seconds by default if not specified. The value of the *timeout-value* parameter ranges from 1 to 60 seconds. It is 1 second by default if not specified.

The value of the *timeout-value* parameter must be smaller than or equal to that of the *interval-value* parameter. The value of the *retry-value* parameter ranges from 1 to 60. It is 1 by default if not specified. The value of the p*riority* parameter ranges from 1 to 255. It is 10 seconds by default if not specified. A VRRP IPv6 link-local IP address is preferred as the tracked IP address of an IPv6 VRRP group. If you set the tracked IP address to a link-local IP address, you must also set the specified interface.

Setting the Periodic Learning of VRRP Advertisement Packets

If the periodic learning function is enabled on the local router that is a VRRP backup router, the local router learns a VRRP advertisement interval from VRRP advertisement packets sent by the master router and calculates a VRRP master invalidity interval using the learned VRRP advertisement interval instead of the VRRP advertisement interval set on the local router itself. This command enables a backup router to synchronize the VRRP advertisement interval locally set on itself to the VRRP advertisement interval of the master router.

Command	Function
Ruijie(config-if)# **vrrp** *group* **timers learn**	Enables the periodic learning of IPv4 VRRP advertisement packets.
Ruijie(config-if)# **no vrrp** *group* **timers learn**	Disables the periodic learning of IPv4 VRRP advertisement packets.
Or:	
Ruijie(config-if)# **vrrp ipv6** *group* **timers learn**	Enables the periodic learning of IPv6 VRRP advertisement packets.
Ruijie(config-if)# **no vrrp ipv6** *group* **timers learn**	Disables the periodic learning of IPv6 VRRP advertisement packets.

By default, the periodic learning function is disabled for a VRRP group.

Note

If the advertisement interval that a VRRP backup router learns from a received VRRP advertisement packet is inconsistent with the VRRP advertisement interval locally set on the VRRP backup router and the periodic learning function is disabled on the VRRP backup router, the VRRP backup router discards the VRRP advertisement packet. Otherwise, the VRRP backup router receives the VRRP advertisement packet and calculates a VRRP master invalidity interval using the advertisement interval carried in the VRRP advertisement packet.

Setting the Description String of the VRRP Group on the Router

You can set a description string for a VRRP group to distinguish it from other VRRP groups.

Command	Function
Ruijie(config-if)# **vrrp** *group* **description** *text*	Sets the description string of an IPv4 VRRP group.
Ruijie(config-if)# **no vrrp** *group* **description**	Cancels the description string set for an IPv4 VRRP group.
Or:	
Ruijie(config-if)# **vrrp ipv6** *group* **description** *text*	Sets the description string of an IPv6 VRRP group.
Ruijie(config-if)# **no vrrp ipv6** *group* **description**	Cancels the description string set for an IPv6 VRRP group.

By default, no description string is set for a VRRP group. The description string of a VRRP group consists of at most 80 bytes.

Note

If the description string of a VRRP group contains blanks, the quotation mark (") must be used to identify the description string.

Setting the Start Delay of the VRRP Group

You can set the start delay of a VRRP group on a certain interface. The system supports two types of delay: system start delay and interface activity delay, which can be configured separately or together.

In non-preemption mode, a router with a higher VRRP group priority does not preempt the master router in the same VRRP group when it is started. In some cases, however, a router newly started preempts other routers to become the VRRP master router, even if it is set to the non-preemption mode. This is because the VRRP group on the interface does not receive the VRRP advertisement packet from the master router in the same VRRP group in time when the router is started or the interface becomes active.

To resolve the preceding problem, you can run a command to configure a start delay for the VRRP group. Then the VRRP group on the interface waits for a certain time before being started when the router is started or the interface becomes active, so that the non-preemption configuration takes effect.

If a VRRP advertisement packet is received on the interface after the start delay is set, the start delay is canceled and VRRP is immediately started on the interface.

Command	Function
Ruijie(config-if)# **vrrp delay** { **minimum** *min-seconds* | **reload** *reload-seconds* }	Sets the start delay of the VRRP group on the interface.
Ruijie(config-if)# **no vrrp delay**	Cancels the start delay set for the VRRP group on the interface.

By default, no start delay is configured for the VRRP group on an interface. Both the system start delay and the interface activity delay as mentioned previously range from 0 to 60 seconds. After this command is configured on an interface, the configurations apply to both IPv4 VRRP and IPv6 VRRP groups on the interface.

Setting the IPv4 VRRP Version

You can set the version of IPv4 VRRP to VRRPv2 or VRRPv3. By default, VRRPv2 is applied.

Command	Function
Ruijie(config-if)# **vrrp** *group* **version** { **2 | 3** }	Sets the IPv4 VRRP version.
Ruijie(config-if)# **no vrrp** *group* **version**	Uses the VRRPv2 by default.

Monitoring and Maintenance of VRRP

The **show vrrp**, **show ipv6 vrrp**, and **debug vrrp** commands are available for monitoring and maintaining VRRP. You can run the **show vrrp** command to check the IPv4 VRRP status of the local router, the **show ipv6 vrrp** command to check the IPv6 VRRP status of the local router, and the **debug vrrp** command to check VRRP information, such as status changes to a VRRP group, VRRP advertisement transmitting/receiving, and VRRP events.

show vrrp

Run the following **show vrrp** commands to check the IPv4 VRRP status of the local router:

Command	Function
Ruijie# **show** [**ipv6] vrrp** [**brief** | *group]*	Displays the IPv4 or IPv6 VRRP status of the local router.

Ruijie# **show** [**ipv6**] **vrrp interface** *type number* [**brief**]	Displays the IPv4 or IPv6 VRRP status of a specific network interface.

Command examples:

3) show [ipv6] vrrp

```
Ruijie# show vrrp
FastEthernet 0/0 - Group 1
State is Backup
Virtual IP address is 192.168.201.1 configured
Virtual MAC address is 0000.5e00.0101
Advertisement interval is 3 sec
Preemption is enabled
min delay is 0 sec
Priority is 100
Master Router is 192.168.201.213 , priority is 120
Master Advertisement interval is 3 sec
Master Down interval is 10.82 sec
FastEthernet 0/0 - Group 2
State is Master
Virtual IP address is 192.168.201.2 configured
Virtual MAC address is 0000.5e00.0102
Advertisement interval is 3 sec
Preemption is enabled
min delay is 0 sec
Priority is 120
Master Router is 192.168.201.217 (local), priority is 120
Master Advertisement interval is 3 sec
Master Down interval is 10.59 sec
Ruijie#show ipv6 vrrp
GigabitEthernet 0/13 - Group 1
  State is Master
  Virtual IPv6 address is as follows:
    FE80::2
    1::2
  Virtual MAC address is 0000.5e00.0201
  Advertisement interval is 1 sec
  Preemption is enabled
    min delay is 0 sec
  Priority is 100
  Master Router is FE80::1 (local), priority is 100
  Master Advertisement interval is 1 sec
  Master Down interval is 3.60 sec
```

The command outputs include the following information:

- Names of Ethernet interfaces where IPv4/IPv6 VRRP groups are configured

- ID, status, priority, preemption mode, VRRP advertisement interval, virtual IP address, and virtual MAC address of each VRRP group configured on the interfaces
- IP address, priority, advertisement interval, and invalidity interval of the master router in each VRRP group
- Tracked interface and priority change metric of each VRRP group

4) show [ipv6] vrrp brief

```
Ruijie# show vrrp brief
Interface       Grp  Pri  Time  Own  Pre  State   Master addr     Group addr
FastEthernet 0/0  1   100  3.60  -    P   Backup  192.168.201.213  192.168.201.1
FastEthernet 0/0  2   120  3.53  -    P   Master  192.168.201.217  192.168.201.2
Ruijie#show ipv6 vrrp brief
Interface           Grp  Pri  timer  Own  Pre  State   Master addr   Group addr
GigabitEthernet 0/13  1   100  3.60   -    P    Master  FE80::1      FE80::2
```

- The command outputs include the following information:
- Names of Ethernet interfaces where IPv4/IPv6 VRRP groups are configured
- ID, status, priority, preemption mode, and virtual IP address of each VRRP group configured on the interfaces
- IP address of the mater router in each VRRP group

5) show [ipv6] vrrp interface

```
Ruijie# show vrrp interface FastEthernet 0/0
FastEthernet 0/0 - Group 1
State is Backup
Virtual IP address is 192.168.201.1 configured
Virtual MAC address is 0000.5e00.0101
Advertisement interval is 3 sec
Preemption is enabled
VRRP standard version is V3
min delay is 0 sec
Priority is 100
Master Router is 192.168.201.213 , priority is 120
Master Advertisement interval is 3 sec
Master Down interval is 10.82 sec
FastEthernet 0/0 - Group 2
State is Master
Virtual IP address is 192.168.201.2 configured
Virtual MAC address is 0000.5e00.0102
Advertisement interval is 3 sec
Preemption is enabled
min delay is 0 sec
Priority is 120
Master Router is 192.168.201.217 (local), priority is 120
Master Advertisement interval is 3 sec
Master Down interval is 10.59 sec
Ruijie#
Ruijie#show ipv6 vrrp inter gig 0/13
GigabitEthernet 0/13 - Group 1
  State is Master
```

```
 Virtual IPv6 address is as follows:
   FE80::2
   1::2
 Virtual MAC address is 0000.5e00.0201
 Advertisement interval is 1 sec
 Preemption is enabled
   min delay is 0 sec
 Priority is 100
 Master Router is FE80::1 (local), priority is 100
 Master Advertisement interval is 1 sec
 Master Down interval is 3.60 sec
```

The command outputs include the following information:

- Names of Ethernet interfaces where IPv4/IPv6 VRRP groups are configured
- ID, status, priority, preemption mode, VRRP advertisement interval, virtual IP address, and virtual MAC address of each VRRP group configured on the interfaces
- IP address, priority, advertisement interval, and invalidity interval of the master router in each VRRP group
- Tracked interface and priority change metric of each VRRP group

Debug vrrp

You can run the following **debug [ipv6] vrrp** commands to enable or disable VRRP debugging on the local router:

Command	Function
Ruijie# **debug [ipv6] vrrp errors**	Enables VRRP error debugging.
Ruijie# **no debug** [**ipv6**] **vrrp errors**	Disables VRRP error debugging.
Ruijie# **debug [ipv6] vrrp events**	Enables VRRP event debugging.
Ruijie# **no debug** [**ipv6**] **vrrp events**	Disables VRRP event debugging.
Ruijie# **debug [ipv6] vrrp packets**	Enables VRRP packet debugging.
Ruijie# **no debug** [**ipv6**] **vrrp packets**	Disables VRRP packet debugging.
Ruijie# **debug [ipv6] vrrp state**	Enables VRRP status debugging..
Ruijie# **no debug** [**ipv6**] **vrrp state**	Disables VRRP status debugging.
Ruijie# **debug [ipv6] vrrp**	Enables VRRP debugging.
Ruijie# **no debug** [**ipv6**] **vrrp**	Disables VRRP debugging.

Command examples:

1) **debug [ipv6] vrrp**

```
Ruijie# debug vrrp
Ruijie#
%VRRP-6-STATECHANGE: FastEthernet 0/0 IPv4 VRRP Grp 1 state Master -> Backup
VRRP: IPv4 VRRP Grp 1 Advertisement from 192.168.201.213 has invalid virtual address 192.168.1.1
VRRP: IPv4 VRRP Grp 1 on interface Gi0/13 is sending IPv4 VRRP V2 advertisement checksum a352.
Ruijie# debug ipv6 vrrp
Ruijie#
VRRP: IPv6 VRRP Grp 1 Event - Advert higher or equal priority
%VRRP-6-STATECHANGE: FastEthernet 0/0 IPv6 VRRP  Grp 1 state Backup -> Master
```

```
Ruijie#
VRRP: IPv6 VRRP Grp 1 on interface Gi0/13 is sending IPv6 VRRP v3 advertisement checksum 6de3.
```

The **debug [ipv6] vrrp** command is equivalent to a combination of commands **debug [ipv6] vrrp errors**, **debug [ipv6] vrrp events**, **debug [ipv6] vrrp packets**, and **debug [ipv6] vrrp state**.

2) debug [ipv6] vrrp errors

```
Ruijie# debug vrrp errors
Ruijie#
VRRP: IPv4 VRRP Grp 1 Advertisement from 192.168.1.1 has wrong checksum.
VRRP: IPv4 VRRP Grp 1 Advertisement from 192.168.1.1 has wrong checksum.
VRRP: IPv4 VRRP Grp 1 Advertisement from 192.168.1.1 has wrong checksum.
```

The preceding information indicates that the local router has received VRRP advertisement packets that contain checksum errors for IPv4 VRRP group 1 from 192.168.1.1.

```
Ruijie# debug ipv6 vrrp errors
Ruijie#
VRRP: IPv6 VRRP Grp 1 Advertisement from FE80::2D0:F8FF:FE22:DE00 has different IP address.
VRRP: IPv6 VRRP Grp 1 Advertisement from FE80::2D0:F8FF:FE22:DE00 has different IP address.
VRRP: IPv6 VRRP Grp 1 Advertisement from FE80::2D0:F8FF:FE22:DE00 has different IP address.
```

The preceding information indicates that the local router has received VRRP advertisement packets that carry different IPv6 group addresses for the same IPv6 VRRP group.

3) debug [ipv6] vrrp events

```
Ruijie# debug vrrp events
Ruijie#
VRRP: IPv4 VRRP Grp 1 Event - Advert higher or equal priority
VRRP: IPv4 VRRP Grp 1 Event - Advert higher or equal priority
Ruijie# debug ipv6 vrrp events
VRRP: IPv6 VRRP Grp 1 Event - Advert higher or equal priority
Ruijie#
```

The preceding information indicates that the local router has received VRRP advertisement packets with a priority higher than or equal to the local priority for local IPv4 VRRP and IPv6 VRRP groups.

4) debug [ipv6] vrrp packets

```
Ruijie# debug vrrp packets
Ruijie#
VRRP: IPv4 VRRP Grp 1 on interface Gi0/13 is sending IPv4 VRRP V2 advertisement checksum a352.
VRRP: IPv4 VRRP Grp 1 on interface Gi0/13 is sending IPv4 VRRP V2 advertisement checksum a352.
Ruijie# debug ipv6 vrrp packets
VRRP: IPv6 VRRP Grp 1 on interface Gi0/13 is sending IPv6 VRRP v3 advertisement checksum 6de3.
VRRP: IPv6 VRRP Grp 1 on interface Gi0/13 is sending IPv6 VRRP v3 advertisement checksum 6de3.
```

The preceding information indicates that local IPv4 VRRP group 1 and local IPv6 VRRP group 1 are sending VRRP advertisement packets.

```
Ruijie# debug vrrp packets
```

```
Ruijie#
VRRP: IPv4 VRRP Grp 1 on interface Gi0/13 received ipv4 v2 advertisement priority 100, source
192.168.1.1.
Ruijie# debug ipv6 vrrp packets

VRRP: IPv6 VRRP Grp 1 on interface Gi0/13 received ipv6 v3 advertisement priority 100, source
FE80::1.
```

The preceding information indicates that the local router has received a VRRP advertisement packet with the priority of 100 for IPv4 VRRP group 1 from 192.168.1.1 and also a VRRP advertisement for IPv6 VRRP group 1 from fe80::1.

5) debug [ipv6] vrrp state

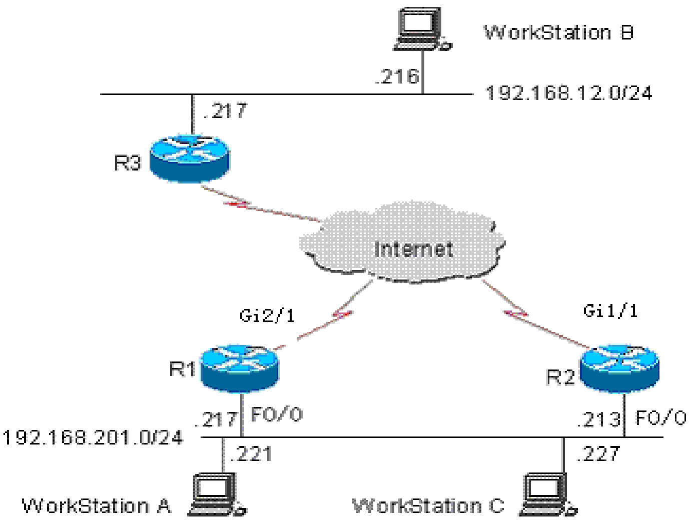
```
Ruijie# debug vrrp state
Ruijie#
VRRP: IPv4 VRRP Grp 1 add primary virtual IP, startup
Ruijie# debug ipv6 vrrp state
VRRP: IPv6 VRRP Grp 1 add primary virtual IP, startup.
```

The preceding information indicates that both the IPv4 VRRP group and the IPv6 VRRP group on the interface FastEthernet 0/0 are configured with a primary IP address and started.

Example of Configuring an IPv4 VRRP Group

As shown in Figure 4, a VRRP group is configured on routers R1 and R2 to provide the VRRP service for the internal network segment 192.168.201.0/24, whereas only the common routing function instead of any VRRP group is enabled on R3. This example shows VRRP-related configurations on routers R1 and R2 only.

Figure 4 VRRP network topology



In the following example, the configuration of router R3 is invariable. The following shows configurations on R3:

```
!
!
hostname "R3"
!
```

```
!
!
interface FastEthernet 0/0
/* The no switchport command needs to be run on a switch only*/
no switchport
ip address 192.168.12.217 255.255.255.0
!
interface GigabitEthernet 1/1
/* The no switchport command needs to be run on a switch only*/
no switchport
ip address 60.154.101.5 255.255.255.0
!
interface GigabitEthernet 2/1
/* The no switchport command needs to be run on a switch only*/
no switchport
ip address 202.101.90.61 255.255.255.0
!
router ospf
network 202.101.90.0 0.0.0.255 area 10
network 192.168.12.0 0.0.0.255 area 10
network 60.154.101.0 0.0.0.255 area 10
!
!
!
end
```

Example of Configuring a VRRP Group

Devices are connected, as shown in Figure 1-4. In this example, a workstation group (192.168.201.0/24) uses a VRRP group formed by routers R1 and R2. Its gateway is set to the virtual router IP address 192.168.201.1 of the VRRP group, so that the workstation group can access a remote workstation group whose network address is 192.168.12.0/24 through the virtual router 192.168.201.1. Here, R1 is set as the master router of the VRRP group. In normal cases, R1 provides the gateway (192.168.201.1) function. If R1 is unreachable because it is shut down or faulty, R2 takes the place of R1 to provide the gateway function. Below are related configurations on R1 and R2.

Configurations on R1:

```
!
!
hostname "R1"
!
!
interface FastEthernet 0/0
ip address 192.168.201.217 255.255.255.0
vrrp 1 priority 120
vrrp 1 version 3
vrrp 1 timers advertise 3
vrrp 1 ip 192.168.201.1
!
interface GigabitEthernet 2/1
ip address 202.101.90.63 255.255.255.0
!
router ospf
network 202.101.90.0 0.0.0.255 area 10
network 192.168.201.0 0.0.0.255 area 10
!
```

Configurations on R2:

```
!
hostname "R2"
!
interface FastEthernet 0/0
ip address 192.168.201.213 255.255.255.0
vrrp 1 ip 192.168.201.1
 vrrp 1 version 3
vrrp 1 timers advertise 3
!
interface GigabitEthernet 1/1
/* The no switchport command needs to be run on a switch only*/
no switchport
ip address 60.154.101.3 255.255.255.0
!
!
router ospf
network 60.154.101.0 0.0.0.255 area 10
network 192.168.201.0 0.0.0.255 area 10
!
!
end
```

As can be seen, R1 and R2 belong to IPv4 VRRP group 1. Both routers use VRRPv3, point to the same virtual router IP address 192.168.201.1, and work in VRRP preemption mode. Since the priority of R1 in the IPv4 VRRP group is 120 but that of R2 is the default value 100, R1 works as the master router of the IPv4 VRRP group in normal cases.

Example of Configuring the Tracked Interface of an IPv4 VRRP Group

Devices are connected, as shown in Figure 4. In this example, a workstation group (192.168.201.0/24) uses a VRRP group formed by routers R1 and R2. Its gateway is set to the virtual router IP address 192.168.201.1 of the VRRP group, so that the workstation group can access a remote workstation group whose network address is 192.168.12.0/24 through the virtual router 192.168.201.1. Here, R1 is set as the master router of the VRRP group. Different from the example as described previously for configuring a single VRRP group, a VRRP tracked interface (GigabitEthernet 2/1) is set on R1. In normal cases, R1 provides the virtual gateway (192.168.201.1) function. If R1 is unreachable because it is shut down or faulty, R2 takes the place of R1 to provide the virtual gateway function. In particular, when the interface GigabitEthernet

2/1 on R1 to connect to a wide area network (WAN) is unavailable, R1 decreases its VRRP group priority based on settings, so that R2 has a chance to become the master router and provide the virtual gateway function. If the interface GigabitEthernet 2/1 on R1 is recovered later, R1 restores its own VRRP group priority and then become the master router to provide the virtual gateway function. Below are related configurations on R1 and R2.

Configurations on R1:

```
!
!
hostname "R1"
!
!
interface FastEthernet 0/0
ip address 192.168.201.217 255.255.255.0
vrrp 1 priority 120
vrrp 1 timers advertise 3
vrrp 1 ip 192.168.201.1
vrrp 1 track GigabitEthernet 2/1 30
!

interface GigabitEthernet 2/1
ip address 202.101.90.63 255.255.255.0
!
router ospf
network 202.101.90.0 0.0.0.255 area 10
network 192.168.201.0 0.0.0.255 area 10
!
!
end
```

Configurations on R2:

```
!
!
hostname "R2"
!
interface FastEthernet 0/0
ip address 192.168.201.213 255.255.255.0
vrrp 1 ip 192.168.201.1
vrrp 1 timers advertise 3
!
interface GigabitEthernet 1/1
ip address 60.154.101.3 255.255.255.0
!
router ospf
network 60.154.101.0 0.0.0.255 area 10
network 192.168.201.0 0.0.0.255 area 10
```

```
!
!
end
```

As can be seen, R1 and R2 belong to VRRP group 1. Both routers use the same VRRP group authentication mode (no-authentication), point to the same virtual router IP address 192.168.201.1, and work in VRRP preemption mode. The VRRP advertisement interval is set to three seconds on both R1 and R2. Since the priority of R1 in the VRRP group is 120 but that of R2 is the default 100, R1 works as the master router of the VRRP group in normal cases. If R1 detects that its interface GigabitEthernet 2/1 to the WAN is unavailable, it decreases its VRRP group priority by 30 to 90, so that R2 becomes the master router. If R1 detects later that its interface GigabitEthernet 2/1 to the WAN is available again, it increases its own VRRP group priority by 30 to 120, so that R1 again becomes the master router.

Configuring Multiple IPv4 VRRP Groups

Multiple VRRP groups can be configured on one Ethernet interface to implement load balancing and backup one another to provide more reliable and stable network services.

Devices are connected, as shown in Figure 4. In this example, a workstation group (192.168.201.0/24) uses two VRRP groups formed by routers R1 and R2. The gateway of some workstations such as workstation A is set to the virtual IP address 192.168.201.1 of VRRP group 1, and that of the rest workstations such as workstation C is set to the virtual IP address 192.168.201.2 of VRRP group 2. R1 serves as the master router of VRRP group 2 and the backup router of VRRP group 1, whereas R2 serves as the master router of VRRP group 1 and the backup router of VRRP group 2. Below are related configurations on R1 and R2.

Configurations on R1:

```
!
!
hostname "R1"
!
interface FastEthernet 0/0
ip address 192.168.201.217 255.255.255.0
vrrp 1 timers advertise 3
vrrp 1 ip 192.168.201.1
vrrp 2 priority 120
vrrp 2 timers advertise 3
vrrp 2 ip 192.168.201.2
vrrp 2 track GigabitEthernet 2/1 30
!
interface GigabitEthernet 2/1
ip address 202.101.90.63 255.255.255.0
!
router ospf
network 202.101.90.0 0.0.0.255 area 10
network 192.168.201.0 0.0.0.255 area 10
!
!
end
```

Configurations on R2:

```
!
!
hostname "R2"
!
interface FastEthernet 0/0
ip address 192.168.201.213 255.255.255.0
vrrp 1 ip 192.168.201.1
vrrp 1 timers advertise 3
vrrp 1 priority 120
vrrp 2 ip 192.168.201.2
vrrp 2 timers advertise 3
!
interface GigabitEthernet 1/1
ip address 60.154.101.3 255.255.255.0
!
router ospf
network 60.154.101.0 0.0.0.255 area 10
network 192.168.201.0 0.0.0.255 area 10
!
!
!
end
```

As can be seen, R1 and R2 back up each other. They serve as the master router in VRRP group 1 or 2 to provide different virtual gateways.

Example of Configuring an IPv6 VRRP Group

Configuring a VRRP Group

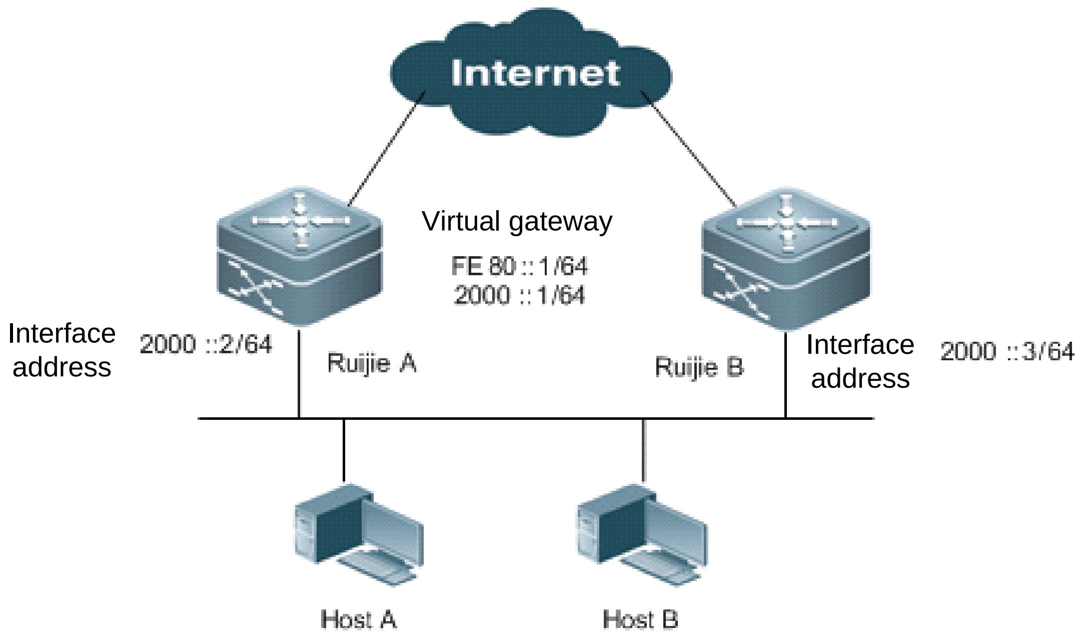
Networking Requirements

This configuration instance is applicable to both switches and routers.

- Hosts A and B access the Internet through a gateway. The default gateway is 2000::1/64 on both hosts.
- Ruijie A and Ruijie B are two routers that form IPv6 VRRP group 1. The virtual addresses are 2000::1/64 and FE80::1.
- When Ruijie A works properly, the packets of Host A to the Internet are forwarded by Ruijie A. When Ruijie A fails, the packets of Host A to the Internet are forwarded by Ruijie B.

Networking Topology

Figure 5 Network topology of the example for configuring an IPv6 VRRP group



Configuration Steps

Configurations on Ruijie A:

# Configure an IPv6 address on an interface to enable the IPv6 service on the interface.

```
interface FastEthernet 0/1
```

/* The **no switchport** command needs to be run on a switch only*/

```
no switchport
ipv6 address 2000::2/64
!
```

# Create IPv6 VRRP group 1 and configure virtual IPv6 addresses FE80::1 and 2000::1.

```
interface FastEthernet 0/0
vrrp 1 ipv6 FE80::1
vrrp 1 ipv6 2000::1
```

# Change the priority of IPv6 VRRP group 1 to 120.

```
vrrp ipv6 1 priority 120
```

# Change the advertisement interval of IPv6 VRRP group 1 to 3s.

```
vrrp ipv6 1 timers advertise 3
```

# Set the Accept_Mode of the IPv6 VRRP group.

```
vrrp ipv6 1 accept_mode
!
```

Configurations on Ruijie B:

# Create IPv6 VRRP group 1 and configure virtual IPv6 addresses FE80::1 and 2000::1.

```
interface FastEthernet 0/1
```

/* The **no switchport** command needs to be run on a switch only*/

```
no switchport
ipv6 address 2000::3/64
!
```

# Create IPv6 VRRP group 1 and configure virtual IPv6 addresses FE80::1 and 2000::1.

```
interface FastEthernet 0/0
vrrp 1 ipv6 FE80::1
vrrp 1 ipv6 2000::1
```

# Change the priority of IPv6 VRRP group 1 to 120.

```
vrrp ipv6 1 priority 100
```

# Change the advertisement interval of IPv6 VRRP group 1 to 3s.

```
vrrp ipv6 1 timers advertise 3
```

# Set the Accept_Mode of the IPv6 VRRP group.

```
vrrp ipv6 1 accept_mode
```

As can be seen, Ruijie A and Ruijie B belong to IPv6 VRRP group 1, point to the same virtual router IPv6 address (2000::1), and work in VRRP preemption mode. Since the priority of Ruijie A in the IPv6 VRRP group is 120 and that of Ruijie B is the default 100, Ruijie A works as the master router of the IPv6 VRRP group in normal cases.

Verification

Run the **show ipv6 vrrp 1** command to check VRRP configuration information after the configuration is complete.

# Show configurations on Ruijie A

```
Ruijie#show ipv6 vrrp 1
FastEthernet 0/1 - Group 1
  State is Master
  Virtual IPv6 address is as follows:
FE80::1
2000::1
  Virtual MAC address is 0000.5e00.0201
  Advertisement interval is 3 sec
  Accept_Mode is enabled
  Preemption is enabled
    min delay is 0 sec
  Priority is 120
  Master Router is FE80::1234 (local), priority is 120
  Master Advertisement interval is 3 sec
  Master Down interval is 10.59 sec
```

# Show configurations on Ruijie B

```
Ruijie#show ipv6 vrrp 1
FastEthernet 0/1 - Group 1
  State is Backup
  Virtual IPv6 address is as follow:
FE80::1
2000::1
  Virtual MAC address is 0000.5e00.0201
  Advertisement interval is 3 sec
  Accept_Mode is enabled
  Preemption is enabled
    min delay is 0 sec
  Priority is 100
  Master Router is FE80::1234, priority is 120
  Master Advertisement interval is 3 sec
  Master Down interval is 10.82 sec
```

Example of Configuring the Tracked Interface of an IPv6 VRRP Group

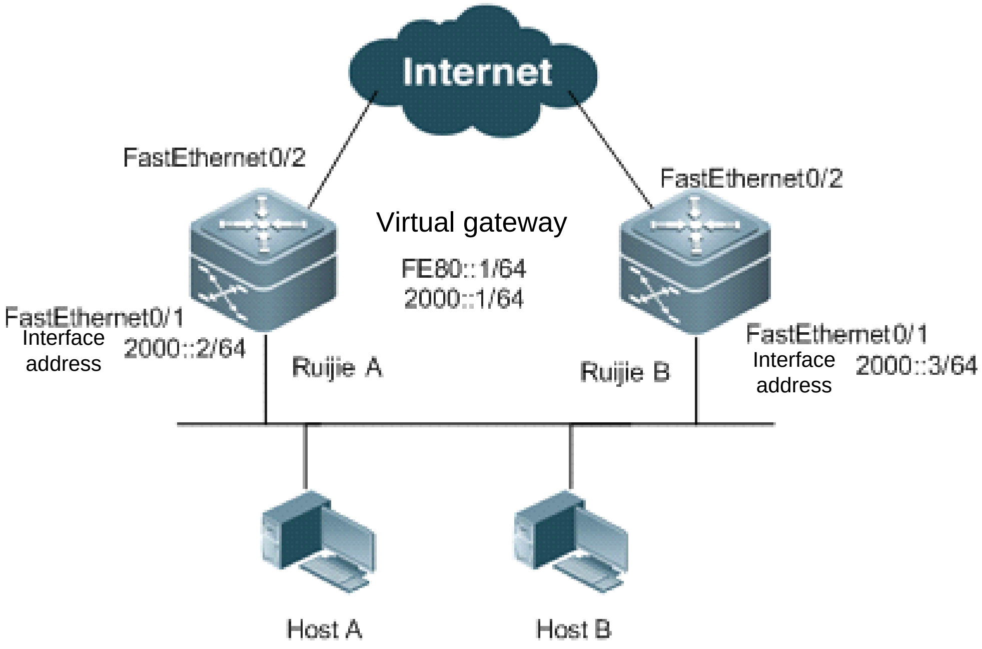
Networking Requirements

This configuration instance is applicable to both switches and routers.

- Host A and Host B access the Internet through a gateway. The default gateway is 2000::1/64 on both hosts.
- Ruijie A and Ruijie B are two routers that form IPv6 VRRP group 1. The virtual addresses are 2000::1/64 and FE80::1.
- Ruijie A tracks the interface FastEthernet 0/2 to the Internet. When the interface FastEthernet 0/2 is unavailable, Ruijie A decreases its VRRP group priority so that Ruijie B serves as the master router to provide the gateway function.

Networking Topology

Figure 6 Network topology of the example for configuring the tracked interface of an IPv6 VRRP group



Configuration Steps

Configurations on Ruijie A:

# Configure an IPv6 address on an interface to enable the IPv6 service on the interface.

```
interface FastEthernet 0/0
```

/* The **no switchport** command needs to be run on a switch only*/

```
no switchport
ipv6 address 2000::2/64
!
```

# Create IPv6 VRRP group 1 and configure virtual IPv6 addresses FE80::1 and 2000::1.

```
interface FastEthernet 0/0
vrrp 1 ipv6 FE80::1
vrrp 1 ipv6 2000::1
!
```

# Change the priority of IPv6 VRRP group 1 to 120.

```
vrrp ipv6 1 priority 120
!
```

# Change the advertisement interval of IPv6 VRRP group 1 to 3s.

```
vrrp ipv6 1 timers advertise 3
!
```

# Configure the tracked interface FastEthernet 0/2 for IPv6 VRRP group 1.

```
vrrp ipv6 1 track FastEthernet 0/2 50
```

# Set the Accept_Mode of the IPv6 VRRP group.

```
vrrp ipv6 1 accept_mode
```

Configurations on Ruijie B:

# Create IPv6 VRRP group 1 and configure virtual IPv6 addresses FE80::1 and 2000::1.

```
interface FastEthernet 0/0
```

/* The **no switchport** command needs to be run on a switch only*/

```
no switchport
ipv6 address 2000::3/64
!
```

# Create IPv6 VRRP group 1 and configure virtual IPv6 addresses FE80::1 and 2000::1.

```
interface FastEthernet 0/0
vrrp 1 ipv6 FE80::1
vrrp 1 ipv6 2000::1
```

# Change the priority of IPv6 VRRP group 1 to 100.

```
vrrp ipv6 1 priority 100
!
```

# Change the advertisement interval of IPv6 VRRP group 1 to 3s.

```
vrrp ipv6 1 timers advertise 3
```

# Set the Accept_Mode of the IPv6 VRRP group.

```
vrrp ipv6 1 accept_mode
```

As can be seen, Ruijie A and Ruijie B belong to IPv6 VRRP group 1, point to the same virtual router IPv6 address (2000::1), and work in IPv6 VRRP preemption mode. Since the priority of Ruijie A in the IPv6 VRRP group is 120 and that of Ruijie B is the default 100, Ruijie A works as the master router of the IPv6 VRRP group in normal cases. If Ruijie A detects that its interface FastEthernet 0/2 is unavailable, it decreases its VRRP group priority by 50 to 70, so that Ruijie B becomes the master router. If Ruijie A detects later that its interface FastEthernet 0/2 is available again, it increases its VRRP group priority by 50 to 120, so that Ruijie A again becomes the master router.

Verification

Run the **show ipv6 vrrp 1** command to check VRRP configuration information after the configuration is complete.

# Show configurations on Ruijie A

```
Ruijie#show ipv6 vrrp 1
FastEthernet 0/1 - Group 1
  State is Master
  Virtual IPv6 address is as follows:
```

```
FE80::1
2000::1
  Virtual MAC address is 0000.5e00.0201
  Advertisement interval is 3 sec
  Accept_Mode is enabled
  Preemption is enabled
    min delay is 0 sec
  Priority is 120
  Master Router is FE80::1234 (local), priority is 120
  Master Advertisement interval is 3 sec
  Master Down interval is 10.59 sec
Tracking state of 1 interface, 1 up:
    up FastEthernet 0/2 priority decrement=50
```

# Show configurations on Ruijie B

```
Ruijie#show ipv6 vrrp 1
FastEthernet 0/1 - Group 1
  State is Backup
  Virtual IPv6 address is as follow:
FE80::1
2000::1
  Virtual MAC address is 0000.5e00.0201
  Advertisement interval is 3 sec
  Accept_Mode is enabled
  Preemption is enabled
    min delay is 0 sec
  Priority is 100
  Master Router is FE80::1234, priority is 120
  Master Advertisement interval is 3 sec
  Master Down interval is 10.82 sec
```

Example of Configuring Multiple IPv6 VRRP Groups

Multiple VRRP groups can be configured on one Ethernet interface to implement load balancing and backup one another to provide more reliable and stable network services.

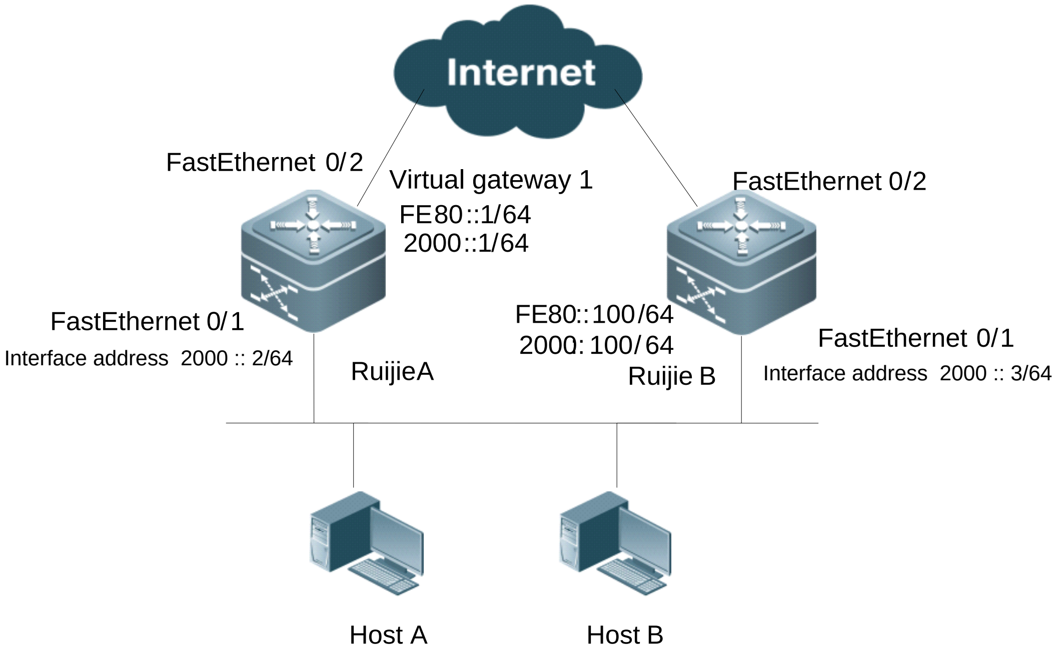
Networking Requirements

This configuration instance is applicable to both switches and routers.

- Host A and Host B access the Internet through gateways. The default gateway for Host A is 2000::1/64, and that for Host B is 2000::100/64.
- Ruijie A and Ruijie B are two routers that form IPv6 VRRP group 1. The virtual addresses are 2000::1/64 and FE80::1.
- Ruijie A and Ruijie B also form IPv6 VRRP group 2. The virtual addresses are 2000::100/64 and FE80::100.
- Ruijie A and Ruijie B serve as gateways to forward traffic and back up each other.

Networking Topology

Figure 7 Network topology of the example for configuring multiple IPv6 VRRP groups



Configuration Steps

Configurations on Ruijie A:

# Configure an IPv6 address on an interface to enable the IPv6 service on the interface.

```
interface FastEthernet 0/0
```

/* The **no switchport** command needs to be run on a switch only*/

```
no switchport
ipv6 address 2000::2/64
!
```

# Create IPv6 VRRP group 1 and configure virtual IPv6 addresses FE80::1 and 2000::1.

```
interface FastEthernet 0/0
vrrp 1 ipv6 FE80::1
vrrp 1 ipv6 2000::1
!
```

# Change the priority of IPv6 VRRP group 1 to 120.

```
vrrp ipv6 1 priority 120
!
```

# Change the advertisement interval of IPv6 VRRP group 1 to 3s.

```
vrrp ipv6 1 timers advertise 3
```

# Set the Accept_Mode of IPv6 VRRP group 1.

```
vrrp ipv6 1 accept_mode
!
```

# Create IPv6 VRRP group 2 and configure virtual IPv6 addresses FE80::100 and 2000::100.

```
vrrp 2 ipv6 FE80::100
vrrp 2 ipv6 2000::100
!
```

# Change the priority of IPv6 VRRP group 2 to 100.

```
vrrp ipv6 2 priority 100
```

# Change the advertisement interval of IPv6 VRRP group 2 to 3s.

```
vrrp ipv6 2 timers advertise 3
```

# Set the Accept_Mode of IPv6 VRRP group 2.

```
vrrp ipv6 2 accept_mode
```

Configurations on Ruijie B:

```
interface FastEthernet 0/0
```

/* The **no switchport** command needs to be run on a switch only*/

```
no switchport
ipv6 address 2000::3/64
!
```

# Create IPv6 VRRP group 1 and configure virtual IPv6 addresses FE80::1 and 2000::1.

```
interface FastEthernet 0/0
vrrp 1 ipv6 FE80::1
vrrp 1 ipv6 2000::1
```

# Change the priority of IPv6 VRRP group 1 to 100.

```
vrrp ipv6 1 priority 100
!
```

# Change the advertisement interval of IPv6 VRRP group 1 to 3s.

```
vrrp ipv6 1 timers advertise 3
```

# Set the Accept_Mode of IPv6 VRRP group 1.

```
vrrp ipv6 1 accept_mode
!
!
```

# Create IPv6 VRRP group 2 and configure virtual IPv6 addresses FE80::100 and 2000::100.

```
vrrp 2 ipv6 FE80::100
vrrp 2 ipv6 2000::100
!
```

# Change the priority of IPv6 VRRP group 2 to 120.

```
vrrp ipv6 2 priority 120
```

# Change the advertisement interval of IPv6 VRRP group 2 to 3s.

```
vrrp ipv6 2 timers advertise 3
!
```

# Set the Accept_Mode of IPv6 VRRP group 2.

```
vrrp ipv6 2 accept_mode
!
```

As can be seen, Ruijie A and Ruijie B belong to IPv6 VRRP group 1, point to the same virtual router IPv6 address (2000::1), and work in IPv6 VRRP preemption mode. Since the priority of Ruijie A in IPv6 VRRP group 1 is 120 and that of Ruijie B is the default 100, Ruijie A works as the master router of IPv6 VRRP group 1 in normal cases. In IPv6 VRRP group 2, however, the priority of Ruijie A is 100 and that of Ruijie B is 120 and IPv6 VRRP group 2 works in preemption mode. Therefore, Ruijie B works as the master router of IPv6 VRRP group 2 in normal cases. For hosts in the same LAN, Host A uses IPv6 VRRP group 1 as the default gateway whereas Host B uses IPv6 VRRP group 2 as the default gateway. Route redundancy is implemented between Ruijie A and Ruijie B, which share LAN traffic and implement load balancing. In this example, default gateways must be manually set on IPv6 hosts to implement load balancing based on IPv6 VRRP groups.

Verification

Run the **show ipv6 vrrp** command to check VRRP configuration information after the configuration is complete.

# Show configurations on Ruijie A

```
Ruijie#show ipv6 vrrp
FastEthernet 0/1 - Group 1
  State is Master
  Virtual IPv6 address is as follows:
FE80::1
2000::1
  Virtual MAC address is 0000.5e00.0201
  Advertisement interval is 3 sec
  Accept_Mode is enabled
  Preemption is enabled
    min delay is 0 sec
  Priority is 120
  Master Router is FE80::1234 (local), priority is 120
  Master Advertisement interval is 3 sec
  Master Down interval is 10.59 sec
FastEthernet 0/1 - Group 2
  State is Backup
  Virtual IPv6 address is as follows:
FE80::100
2000::100
  Virtual MAC address is 0000.5e00.0202
  Advertisement interval is 3 sec
  Accept_Mode is enabled
  Preemption is enabled
    min delay is 0 sec
  Priority is 100
  Master Router is FE80::5678, priority is 120
  Master Advertisement interval is 3 sec
  Master Down interval is 10.82 sec
```

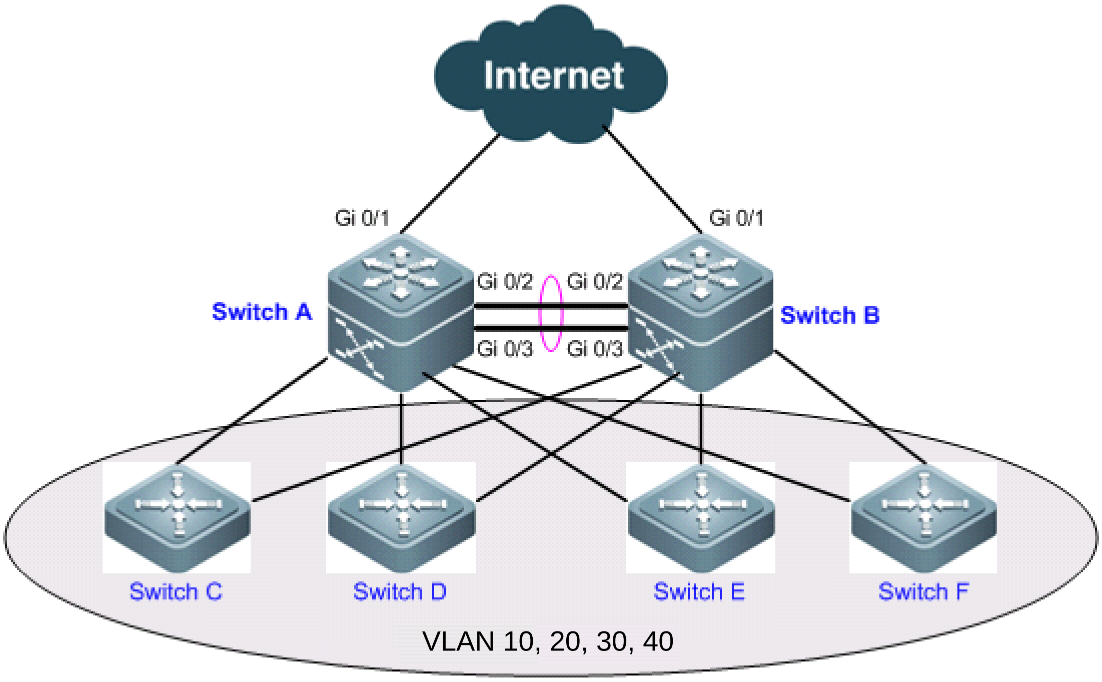
# Show configurations on Ruijie B

```
Ruijie#show ipv6 vrrp 1
FastEthernet 0/1 - Group 1
  State is Backup
  Virtual IPv6 address is as follow:
FE80::1
2000::1
  Virtual MAC address is 0000.5e00.0201
  Advertisement interval is 3 sec
  Accept_Mode is enabled
  Preemption is enabled
    min delay is 0 sec
  Priority is 100
  Master Router is FE80::1234, priority is 120
  Master Advertisement interval is 3 sec
  Master Down interval is 10.82 sec

FastEthernet 0/1 - Group 2
  State is Master
  Virtual IPv6 address is as follows:
FE80::100
2000::100
  Virtual MAC address is 0000.5e00.0202
  Advertisement interval is 3 sec
  Accept_Mode is enabled
  Preemption is enabled
    min delay is 0 sec
  Priority is 120
  Master Router is FE80::5678(local), priority is 120
  Master Advertisement interval is 3 sec
  Master Down interval is 10.59 sec
```

Configuring VRRP+MSTP

Networking Topology

Figure 8 Network topology of the VRRP dual-core solution



Networking Requirements

Figure 8 shows a typical network topology of the dual-core solution. This configuration instance is applicable to switches or switching cards of routers only. Users access Switches C, D, E, and F, which belong to VLAN 10, 20, 30, and 40 respectively. Switches A and B serve as gateways to enable users to communicate with external networks. The specific application requirements are described as follows:

- The Multiple Spanning Tree Protocol (MSTP) runs on devices to back up physical links and avoid loops. Different VLAN packets are forwarded along respective instances to implement layer 2 traffic load balancing.
- VRRP runs on devices to back up gateway routes and share LAN traffic.
- All links from access switches to the master router are monitored. When a link to the master router fails, the backup router immediately takes over the master router to forward data.

Configuration Tips

- Enable the MSTP function on devices (Switches A, B, C, D, E, and F in this example), configure mappings between VLANs and instances (VLANs 10 and 20 map to instance 1, VLANs 30 and 40 map to instance 2, and the rest VLANs map to instance 0 in this example), and set gateways (Switches A and B in this example) as the root bridges of respective instances.
- Add the switch virtual instances (SVIs) of various VLANs to respective VRRP groups, and set the master and backup routers of respective VRRP groups as gateways. The following table shows the specific configurations.

Gateway	VLAN ID	SVI	VRRP Group	Virtual IP Address	Status
Switch A	10	192.168.10.2	VRRP 10	192.168.10.1	Master
Switch B		192.168.10.3			Backup
Switch A	20	192.168.20.2	VRRP 20	192.168.20.1	Master
Switch B		192.168.20.3			Backup
Switch A	30	192.168.30.2	VRRP 30	192.168.30.1	Backup

Gateway	VLAN ID	SVI	VRRP Group	Virtual IP Address	Status
Switch B		192.168.30.3			Master
Switch A	40	192.168.40.2	VRRP 40	192.168.40.1	Backup
Switch B		192.168.40.3			Master

- Set the uplink ports on the master routers of VRRP groups as the tracked interfaces of the master routers. In this example, the tracked interfaces are two ports Gi 0/1 on Switches A and B.

Caution

When setting the tracked interface of a VRRP group, ensure that the value of the *Priority decrement* parameter is larger than the difference between the priority of the master router and the priority of the backup router. The system automatically decreases or increases the priority value of a router according to the status of the tracked interface on the router.

Configuration Steps

In this example, only VRRP+MSTP configurations on Switches A and B are listed. This example does not provide details about how to define VLANs on Switches C, D, E, and F or how to configure MSTP on devices. For details about MSTP configuration, see *MSTP Configuration*.

- Step 1: Create VLANs on devices.

！ Create VLANs 10, 20, 30, and 40 on Switch A:

```
SwitchA#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
SwitchA(config)#vlan range 10,20,30,40
SwitchA(config-vlan-range)#exit
```

！ Configure the same as above on Switch B.

- Step 2: Configure the Multiple Spanning Tree (MST) domain.

！ Configure mappings from VLANs 10 and 20 to instance 1, from VLANs 20 and 30 to instance 2, and from the rest VLANs to instance 0 on Switch A.

```
SwitchA(config)#spanning-tree mst configuration
SwitchA(config-mst)#instance 1 vlan 10,20
%Warning: you must create vlans before configuring instance-vlan relationship
SwitchA(config-mst)#instance 2 vlan 30,40
%Warning: you must create vlans before configuring instance-vlan relationship
SwitchA(config-mst)#exit
```

! Configure the same as above on Switch B.

Step 3: Set Switch A as the root bridges of MST instances 0 and 1, and set Switch B as the root bridge of MST instance 2.

! Set the priority of MST instances 0 and 1 to 4096 and that of MST instance 2 to 8192 on Switch A.

```
SwitchA(config)#spanning-tree mst 0 priority 4096
SwitchA(config)#spanning-tree mst 1 priority 4096
SwitchA(config)#spanning-tree mst 2 priority 8192
```

! Set the priority of MST instances 0 and 1 to 8192 and that of MST instance 2 to 4096 on Switch B.

```
SwitchB(config)#spanning-tree mst 2 priority 4096
SwitchB(config)#spanning-tree mst 0 priority 8192
SwitchB(config)#spanning-tree mst 1 priority 8192
```

- Step 4: Enable MSTP.

! Enable MSTP on Switch A.

```
SwitchA(config)#spanning-tree
Enable spanning-tree.
```

! Configure the same as above on Switch B.

- Step 5: Configure the SVIs of VLANs, add the SVIs to VRRP groups, and set the virtual IP addresses of VRRP groups. For details, see the preceding table.

! Configurations on Switch A:

```
SwitchA(config)#interface vlan 10
SwitchA(config-if-VLAN 10)#ip address 192.168.10.2 255.255.255.0
SwitchA(config-if-VLAN 10)#vrrp 10 ip 192.168.10.1
SwitchA(config-if-VLAN 10)#exit
SwitchA(config)#interface vlan 20
SwitchA(config-if-VLAN 20)#ip address 192.168.20.2 255.255.255.0
SwitchA(config-if-VLAN 20)#vrrp 20 ip 192.168.20.1
SwitchA(config-if-VLAN 20)#exit
SwitchA(config)#interface vlan 30
SwitchA(config-if-VLAN 30)#ip address 192.168.30.2 255.255.255.0
SwitchA(config-if-VLAN 30)#vrrp 30 ip 192.168.30.1
SwitchA(config-if-VLAN 30)#exit
SwitchA(config)#interface vlan 40
SwitchA(config-if-VLAN 40)#ip address 192.168.40.2 255.255.255.0
SwitchA(config-if-VLAN 40)#vrrp 40 ip 192.168.40.1
SwitchA(config-if-VLAN 40)#exit
```

! Configurations on Switch B:

```
SwitchB(config)#interface vlan 10
SwitchB(config-if-VLAN 10)#ip address 192.168.10.3 255.255.255.0
SwitchB(config-if-VLAN 10)#vrrp 10 ip 192.168.10.1
SwitchB(config-if-VLAN 10)#exit
SwitchB(config)#interface vlan 20
SwitchB(config-if-VLAN 20)#ip address 192.168.20.3 255.255.255.0
SwitchB(config-if-VLAN 20)#vrrp 20 ip 192.168.20.1
SwitchB(config-if-VLAN 20)#exit
SwitchB(config)#interface vlan 30
SwitchB(config-if-VLAN 30)#ip address 192.168.30.3 255.255.255.0
SwitchB(config-if-VLAN 30)#vrrp 30 ip 192.168.30.1
SwitchB(config-if-VLAN 30)#exit
SwitchB(config)#interface vlan 40
SwitchB(config-if-VLAN 40)#ip address 192.168.40.3 255.255.255.0
SwitchB(config-if-VLAN 40)#vrrp 40 ip 192.168.40.1
SwitchB(config-if-VLAN 40)#exit
```

- Step 6: Configure the master and backup routers of VRRP groups.

! Raise the priority of VRRP groups 10 and 20 on Switch A to 120, so that Switch A works as the master router of VRRP groups 10 and 20.

```
SwitchA(config)#interface vlan 10
SwitchA(config-if-VLAN 10)#vrrp 10 priority 120
SwitchA(config-if-VLAN 10)#exit
SwitchA(config)#interface vlan 20
SwitchA(config-if-VLAN 20)#vrrp 20 priority 120
SwitchA(config-if-VLAN 20)#exit
```

! Similarly, raise the priority of VRRP groups 30 and 40 on Switch B to 120.

```
SwitchB(config)#interface vlan 30
SwitchB(config-if-VLAN 30)#vrrp 30 priority 120
SwitchB(config-if-VLAN 30)#exit
SwitchB(config)#interface vlan 40
SwitchB(config-if-VLAN 40)#vrrp 40 priority 120
SwitchB(config-if-VLAN 40)#exit
```

- Step 7: Set the uplink ports on the master routers of VRRP groups as the tracked interfaces of VRRP groups. Ensure that the configured tracked interfaces are L3 interfaces.

! Set the port Gi 0/1 on Switch A as a route port and its IP address to 10.10.1.1/24.

```
SwitchA(config)#interface gigabitEthernet 0/1
SwitchA(config-if-GigabitEthernet 0/1)#no switchport
SwitchA(config-if-GigabitEthernet 0/1)#ip address 10.10.1.1 255.255.255.0
SwitchA(config-if-GigabitEthernet 0/1)#exit
```

! Set the port Gi 0/1 on Switch A as the tracked interface of VRRP groups 10 and 20, and *Priority decrement* to 30.

```
SwitchA(config)#interface vlan 10
SwitchA(config-if-VLAN 10)#vrrp 10 track gigabitEthernet 0/1 30
SwitchA(config-if-VLAN 10)#exit
SwitchA(config)#interface vlan 20
SwitchA(config-if-VLAN 20)#vrrp 20 track gigabitEthernet 0/1 30
SwitchA(config-if-VLAN 20)#exit
```

! Set the port Gi 0/1 on Switch B as a route port and its IP address to 10.10.2.1/24.

```
SwitchB(config)#interface gigabitEthernet 0/1
SwitchB(config-if-GigabitEthernet 0/1)#no switchport
SwitchB(config-if-GigabitEthernet 0/1)#ip address 10.10.2.1 255.255.255.0
SwitchB(config-if-GigabitEthernet 0/1)#exit
```

! Set the port Gi 0/1 on Switch B as the tracked interface of VRRP groups 30 and 40, and *interface-priority* to 30.

```
SwitchB(config)#interface vlan 30
SwitchB(config-if-VLAN 30)#vrrp 30 track gigabitEthernet 0/1 30
SwitchB(config-if-VLAN 30)#exit
SwitchB(config)#interface vlan 40
SwitchB(config-if-VLAN 40)#vrrp 40 track gigabitEthernet 0/1 30
SwitchB(config-if-VLAN 40)#exit
```

- Step 8: Set the interconnection ports between the two core devices (Switches A and B) as aggregation ports.

! Configurations on Switch A:

# Set ports Gi 0/2 and Gi 0/3 as aggregation ports, which serve as trunk ports.

```
Ruijie#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Ruijie(config)#interface range gigabitEthernet 0/2-3
Ruijie(config-if-range)#port-group 1
Ruijie(config)#interface aggregateport 1
Ruijie(config-if-AggregatePort 1)#switchport mode trunk
```

! Configure the same as above on Switch B.

Verification

- Step 1: Check configuration information on devices.

! Check configuration information on Switch A.

```
SwitchA#show running-config
!
vlan 10
!
vlan 20
!
vlan 30
!
vlan 40
!
spanning-tree
spanning-tree mst configuration
 instance 0 vlan 1-9, 11-19, 21-29, 31-39, 41-4094
 instance 1 vlan 10, 20
 instance 2 vlan 30, 40
spanning-tree mst 0 priority 4096
spanning-tree mst 1 priority 4096
spanning-tree mst 2 priority 8192
interface GigabitEthernet 0/1
 no switchport
 no ip proxy-arp
 ip address 10.10.1.1 255.255.255.0
```

```
!
interface GigabitEthernet 0/2
 port-group 1
!
interface GigabitEthernet 0/3
 port-group 1
!
interface AggregatePort 1
 switchport mode trunk
!
interface VLAN 10
 no ip proxy-arp
 ip address 192.168.10.2 255.255.255.0
 vrrp 10 priority 120
 vrrp 10 ip 192.168.10.1
 vrrp 10 track GigabitEthernet 0/1 30
!
interface VLAN 20
 no ip proxy-arp
 ip address 192.168.20.2 255.255.255.0
 vrrp 20 priority 120
 vrrp 20 ip 192.168.20.1
 vrrp 20 track GigabitEthernet 0/1 30
!
interface VLAN 30
 no ip proxy-arp
 ip address 192.168.30.2 255.255.255.0
 vrrp 30 ip 192.168.30.1
!
interface VLAN 40
 no ip proxy-arp
 ip address 192.168.40.2 255.255.255.0
 vrrp 40 ip 192.168.40.1
```

! Check configuration information on Switch B.

```
SwitchB#show running-config
!
vlan 10
!
vlan 20
!
vlan 30
!
vlan 40
!
```

```
spanning-tree
spanning-tree mst configuration
 instance 0 vlan 1-9, 11-19, 21-29, 31-39, 41-4094
 instance 1 vlan 10, 20
 instance 2 vlan 30, 40
spanning-tree mst 0 priority 8192
spanning-tree mst 1 priority 8192
spanning-tree mst 2 priority 4096
interface GigabitEthernet 0/1
 no switchport
 no ip proxy-arp
 ip address 10.10.2.1 255.255.255.0
!
interface GigabitEthernet 0/2
 port-group 1!
interface GigabitEthernet 0/3
 port-group 1
!
interface AggregatePort 1
 switchport mode trunk
!
interface VLAN 10
 no ip proxy-arp
 ip address 192.168.10.3 255.255.255.0
 vrrp 10 ip 192.168.10.1
!
interface VLAN 20
 no ip proxy-arp
 ip address 192.168.20.3 255.255.255.0
 vrrp 20 ip 192.168.20.1
!
interface VLAN 30
 no ip proxy-arp
 ip address 192.168.30.3 255.255.255.0
 vrrp 30 priority 120
 vrrp 30 ip 192.168.30.1
 vrrp 30 track GigabitEthernet 0/1 30
!
interface VLAN 40
 no ip proxy-arp
 ip address 192.168.40.3 255.255.255.0
 vrrp 40 priority 120
 vrrp 40 ip 192.168.40.1
 vrrp 40 track GigabitEthernet 0/1 30
```

- Step 2: Check the VRRP status of each device.

! Check the VRRP status of Switch A.

```
SwitchA#show vrrp brief
Interface  Grp  Pri  timer  Own  Pre  State   Master addr   Group addr
VLAN 10    10   120  3      -    P    Master  192.168.10.2  192.168.10.1
VLAN 20    20   120  3      -    P    Master  192.168.20.2  192.168.20.1
VLAN 30    30   100  3      -    P    Backup  192.168.30.3  192.168.30.1
VLAN 40    40   100  3      -    P    Backup  192.168.40.3  192.168.40.1
```

! Check the VRRP status of Switch B.

```
SwitchB#show vrrp brief
Interface  Grp  Pri  timer  Own  Pre  State   Master addr   Group addr
VLAN 10    10   100  3      -    P    Backup  192.168.10.2  192.168.10.1
VLAN 20    20   100  3      -    P    Backup  192.168.20.2  192.168.20.1
VLAN 30    30   120  3      -    P    Master  192.168.30.3  192.168.30.1
VLAN 40    40   120  3      -    P    Master  192.168.40.3  192.168.40.1
```

As can be seen from above, Switch A serves as the master router of VRRP groups 10 and 20 and has a priority of 120 when links are normal, whereas Switch B serves as the backup routers of VRRP groups 10 and 20.

- Step 3: Disconnect the uplink of Switch A, and then check the VRRP status of Switches A and B.

! Check the VRRP status of Switch A.

```
SwitchA#show vrrp brief
Interface  Grp  Pri  timer  Own  Pre  State   Master addr   Group addr
VLAN 10    10   90   3      -    P    Backup  192.168.10.3  192.168.10.1
VLAN 20    20   90   3      -    P    Backup  192.168.20.3  192.168.20.1
VLAN 30    30   100  3      -    P    Backup  192.168.30.3  192.168.30.1
VLAN 40    40   100  3      -    P    Backup  192.168.40.3  192.168.40.1
```

! Check the VRRP status of Switch B.

```
SwitchB#show vrrp brief
Interface  Grp  Pri  timer  Own  Pre  State   Master addr   Group addr
VLAN 10    10   100  3      -    P    Master  192.168.10.3  192.168.10.1
VLAN 20    20   100  3      -    P    Master  192.168.20.3  192.168.20.1
VLAN 30    30   120  3      -    P    Master  192.168.30.3  192.168.30.1
VLAN 40    40   120  3      -    P    Master  192.168.40.3  192.168.40.1
```

As can be seen from above, when the uplink of Switch A fails, the system automatically decreases the priority of VRRP groups 10 and 20 to 90 and changes the VRRP status of Switch A to Backup, so that Switch B becomes the master router of VRRP groups 10 and 20.

Fault Diagnosis and Clearance

You can analyze and clear VRRP faults by checking configuration and debugging information. Below are some common faults and analysis methods:

Symptom

The virtual IPv4/IPv6 address of a VRRP group cannot be pinged.

Analysis

- Ensure that at least one router is active in the VRRP group.
- If the virtual IPv4/IPv6 address cannot be pinged from a device on another network, possibly the fault is caused because a short time is needed for VRRP status switching. Run the **show [ipv6] vrrp** command to check VRRP information and verify the root cause.
- If the virtual IPv4/IPv6 address cannot be pinged from a local network device in the same network segment as the virtual router, check whether the ARP table or neighbor discovery (ND) table on the local network device contains an ARP entry for the virtual IPv4/IPv6 address. If there is no ARP entry for the virtual IPv4/IPv6 address, check network lines.
- If the virtual IPv4/IPv6 address cannot be pinged from a local network device in a network segment that is different from the network segment where the virtual router resides, check whether a route to the virtual IPv4/IPv6 address has been configured on the local network device.

Symptom

Multiple master routers exist in a VRRP group.

Analysis

- Ethernet interfaces on the routers of the VRRP group are set to different VRRP group authentication modes.
- For VRRPv2, Ethernet interfaces on the routers of the VRRP group are set to the same plain text authentication mode but the configured authentication strings are inconsistent.
- The cables of the Ethernet interfaces on the routers of the VRRP group are disconnected, but the routers fail to detect the disconnection.
- The VRRP advertisement intervals configured on routers of the VRRP group are inconsistent, and the periodic learning of VRRP advertisement packets is disabled on the routers.
- Different virtual IPv4/IPv6 addresses are set for routers of the VRRP group.

Configuring Hot Swapping

Understanding Hot Swap

Overview

Hot swap refers to the ability of allowing removing a faulty line card and inserting the standby line card in a high-reliable system without restarting and shutting down the system.

Ruijie designs hot-swap products by complying with the rule of separating software settings from hardware application and separating the existence of line cards from the startup of line cards.

Ruijie employs the install/no install concept so that users can run the **install** command to pre-install and pre-configure a certain type of line card in a slot even if no line card is inserted in the slot. Management software can reserve all configurations of the line card. Whether to configure hardware depends on actual situations.

After users save settings, all pre-configuration information is stored. After routers are reset, this pre-configuration information is still effective.

During normal running of routers, removing and then inserting a line card will not lead to the loss of related hardware settings.

If the actual type of the newly inserted line card is different from the pre-configured type, the inserted line card is not enabled and users need to uninstall the original configuration to enable the line card.

The concept of “Two Hosts” is introduced in RSR30-X series routers. Every RSR30-X series router includes an LPU expansion box like an LPU-4HNM expansion box. One such LPU expansion box is under the management of two PSP30-X hosts with one host as a MASTER and another as SLAVE. MASTER serves as the major manager and SLAVE as the standby manager. When MASTER is faulty, SLAVE manages LPU. In this way, two hosts guarantee the reliability of LPU.

⚠ Caution

Comply with the following rules during hot swap to avoid causing abnormalities of software and hardware or even damaging line cards:

Operation Limits	Description
Remove Line Cards	1. Run the **remove** or **no install** commands, and remove line cards after the commands are executed. Print log similar to the one below to confirm that the **remove** or **no install** commands are fully executed. *May 17 10:24:03: %HOTPLUG-5-DISABLE_LINE_CARD_BEGIN: Begin to disable the line card in slot 1/0. *May 17 10:24:07: %HOTPLUG-5-DISABLE_LINE_CARD_OK: Disable line card in slot 1/0 OK.

	2. Remove one line card only at a time;
	3. Smoothly and securely remove line cards;
	4. During system startup, the line card must not be inserted;
	5. Print log similar to the one below to confirm that line cards are removed.
	`*May 17 10:28:38: %HOTPLUG-5-PULL_LINE_CARD_BEGIN: Begin process PULL event in slot 1/0.`
	`*May 17 10:28:38: %HOTPLUG-5-PULL_LINE_CARD_OK: PULL event in slot 1/0 process OK.`

Insert Line Cards	1. Line cards can only be inserted into slots after the completion of execution of the hot swap commands or previous removal of line cards. 2. Insert one line card only at a time; 3. Smoothly and securely insert line cards; 4. During system startup, the line card must not be inserted; 5. Line cards are only allowed to be inserted again after they are completely removed (slots and line cards are fully disconnected).

Configuring Hot Swap

Hot Swap Configuration Task List

Hot swap can be configured by following the task list below:

Installing and Uninstalling a Line Card Module

The pre-configuration function of Ruijie products allows users to run the **install** command to virtualize a line card module of the specified type (the line card module is not actually inserted in the slot), and then configure the line card module. After the line card module is inserted in the slot, all configuration automatically takes effect.

Command	Function
Ruijie(config)# **install** *slot-num moduletype*	Installs a line card module.
Ruijie(config)# **no install** *slot-num*	Uninstalls a line card module.

Inserting and Removing a Line Card Module

Follow the rules in the "Overview" section to insert and remove a line card.

You do not need to configure a command to insert a line card.

Before removing a line card, run the **remove** command.

Command	Function
Ruijie(config)# **remove** *slot-num*	Removes a line card.

Ruijie(config)# **no remove** *slot-num*	Restores line card configuration.

Note

When you run the **no remove** command, or directly insert the NMX-8E1/CE1 or NMX-4E1/CE1 line card, the communication over the interface that is not added to the fast forwarding group is interrupted for about 30 seconds. Therefore, it is not recommended that you carry out such operation during peak hours of communication.

During system startup, the line card must not be inserted or removed.

A line card can be inserted and removed only after the system is started completely (namely, the console prompts “%SYS-5-WARMSTART:System warmstart.” or “%SYS-5-WARMSTART:System coldstart.”).

Resetting a Line Card Module Hot Swap

Hot swap resetting includes a series of operations: run the **remove** command, remove a line card, run the **no install** command, run the **install** command, and insert the line card. After the hot swap resetting is complete, line card hardware is reset and configuration information on the software is initialized again.

Command	**Function**
Ruijie(config)# **reset** *slot-num*	Resets a line card module hot swappable.

Monitoring and Maintaining Hot Swap

The **show version slots** command is used to display the information of the line card modules in each slot, including line card types configured by users, actual line card types, and hot swap status of line cards.

Command	**Function**

Ruijie# **show version slots, or** Ruijie(config)#**show version slots**	Display information about line-card modules in all slots, including the type of the line cards configured by users, the type of the inserted line cards, the hos swap status of the line cards, and information about MASTER and SLAVE. 3 status of the management board: none- No management board is inserted in the slot; slave- The management board in the slot is SLAVE. MASTER- The management board in the slot is MASTER. 8 status of line cards: ● None: Neither a line card nor a pre-configured line card is in the slot. ● Installed: No line card is in the slot, but a line card is configured. ● running-config: MASTER is downloading configuration for the line card. ● running: The line card is running properly. ● run-remove: The user has run the **remove** command but has not removed the line card. ● conflict: The inserted line card is not the type pre-configured by the user. ● unins-remove: The user has run the **no install** command but has not removed the line card. ● standby: this status only works for the LPU expansion box and the device is SLAVE.

MASTER/SLAVE Switchover of LPU Expansion Boxes

Under the mode of “Two Hosts”, when MASTER is faulty, MASTER/SLAVE switchover will happen automatically. However, manual switchover is demanded in some cases, such as demonstration of the function. So, the **switchover** command is offered to serve this aim.

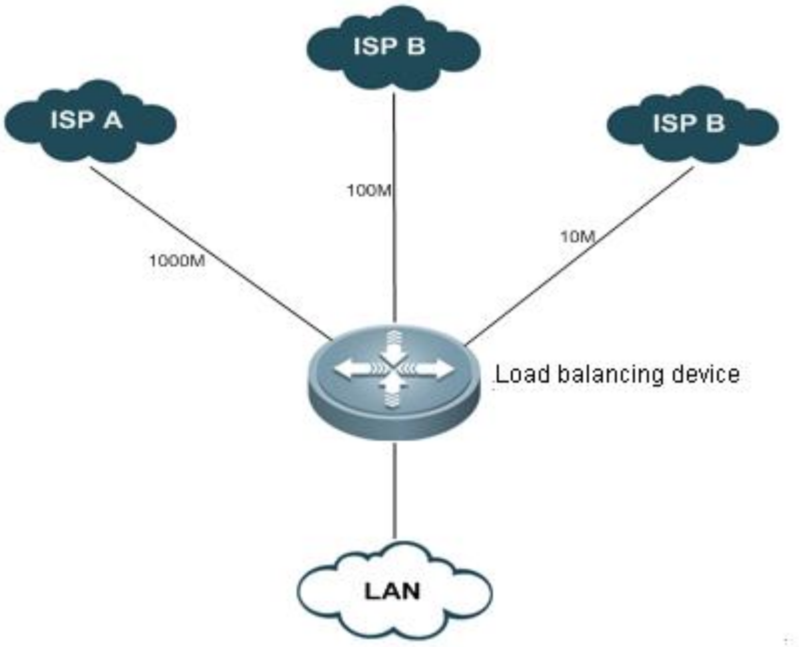
Command	Function

Ruijie(config)#**switchover** *slot-num*	Realizes manual switchover between MASTER and SLAVE.

Configuring Multilink Gateway Load Balancing

Introduction to Multilink Gateway Load Balancing

Fig 1 Typical application diagram of multilink gateway load balancing



Overview

The network gateway is generally connected with two or more ISP links. For example, the gateway of an educational institution will be connected with an education communication link and a Telecom/CNC link; the gateway of a governmental agency may be connected with a Telecom link and a CNC link. Multiple ISP links handles traffic as per certain policy or act as the backup link.

Multilink load balancing allows reasonable flow distribution among multiple links as per certain policy, well improving the utilization efficiency of link resources.

Basic Concept

Link Bandwidth

Link bandwidth is the indicator for measuring available resources, and is different from the transmission rate of physical interface. It is the maximum transmission rate provided by ISP, and generally refers to the inbound bandwidth.

Link Latency

When there are multiple gateways, packets can reach the same destination address through different gateways. The link latency used in load balancing refers to the different response times when packets reach the same destination address through different gateways. It is used to compare the access speed of different gateways, focusing mainly on the difference between links in terms of latency.

Link Load

Link load refers to the current resource utilization rate of the link. It can be calculated by dividing the packet reception rate of physical interface by link bandwidth.

Load Balancing Policy

The policy to share traffic between different links. You can choose load balancing according to bandwidth, access speed, link utilization rate, comprehensive link bandwidth, latency, or load.

Working Principle

In accordance with the load balancing policy selected, the system will calculate the weight of respective links according to link bandwidth, latency and load, and regenerate WCMP routes as per the weight of each link. The subsequent traffic will select the gateway according to the routes generated, thus controlling the traffic handled by each link.

Protocol Specification

NA

Default Configurations

The following table describes the default configurations of multilink gateway load balancing.

Function	Default setting
Configure multilink gateway load balancing	Disabled

Configuring Multilink Gateway Load Balancing

The following section describes how to configure multilink load balancing:

- (Required) Enable/disable multilink load balancing
- (Optional) Configure link bandwidth
- (Optional) Configure link load threshold
- (Optional) Configure load balancing policy
- (Optional) Configure weight base
- Display configurations

Enabling/Disabling Multilink Load Balancing

By default, multilink load balancing function is disabled on the device. Enter privilege mode and execute the following steps to enable multilink load balancing:

Command	Function
Ruijie# **configure terminal**	Enter global configuration mode.
Ruijie(config)# **mllb enable**	Enable multilink load balancing
Ruijie(config)# **no mllb enable**	Disable multilink load balancing

Configuration example:

# Enable global multilink load balancing

```
Ruijie# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# mllb
```

# Disable global multilink load balancing

```
Ruijie# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# no mllb
```

Configuring Link Bandwidth

To configure the bandwidth of interface, execute the following commands in interface configuration mode:

Command	Function
Ruijie# **configure terminal**	Enter global configuration mode.
Ruijie(config)# **interface** *interface-name*	Enter interface configuration mode.
Ruijie(config-if)# **bandwidth** *kilobits*	Configure link bandwidth.
Ruijie(config-if)# **no bandwidth**	Restore link bandwidth to default value.

Configuration example:

# Configure link bandwidth to 1M:

```
Ruijie# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# interface fastEthernet 0/1
Ruijie(config-if)# bandwidth 1000
```

# Remove link bandwidth configurations:

```
Ruijie# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# interface fastEthernet 0/1
Ruijie(config-if)# no bandwidth
```

Configuring Link Load Threshold

If the link load exceeds the threshold configured, the system will then no longer use this link as the gateway link for load balancing. This threshold shall apply to all gateway links. The load threshold will be expressed in percentage (1-100).

To configure link load threshold, execute the following steps:

Command	Function
Ruijie# **configure terminal**	Enter global configuration mode.
Ruijie(config)# **mllb threshold** *percent*	Configure link load threshold to "percent" (integer between 1-100).
Ruijie(config)# **no mllb threshold**	Restore the link load threshold to the default value of 100.

Configuration example:

# Configure link load threshold to 95:

```
Ruijie# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# mllb threshold 95
```

# Restore link load threshold to the default value:

```
Ruijie# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# no mllb threshold
```

Configuring Load Balancing Policy

Load balancing policy provides four different load balancing options for the user. In case of bandwidth based load balancing, the gateway traffic will be shared according to gateway bandwidth; in case of latency based load balancing, the link with shorter latency will handle more traffic and the link with longer latency will handle less traffic; in case of load based load balancing, the traffic will be shared according to the load situation of each gateway link, so that loads are balanced on links; in case of intelligent load balancing, the traffic will be routed by giving comprehensive consideration to bandwidth, latency and load. The user can further configure the weight base of these three factors (see "Configure weight base"), so as to adjust the specific influence of the corresponding factor on flow distribution.

To configure load balancing policy, execute the following steps:

Command	Function
Ruijie# **configure terminal**	Enter global configuration mode.
Ruijie(config)# **mllb policy** { **bandwidth** | **latency** | **load** | **intelligent** }	Specify load balancing policy as bandwidth/latency/load/intelligent policy, with bandwidth being the default policy.
Ruijie(config)# **no mllb policy**	Restore load balancing policy to default setting

Configuration example:

# Configure load balancing policy to bandwidth:

```
Ruijie# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# mllb policy bandwidth
```

Configuring Weight Base

This refers to the corresponding weight base for calculating the link weight as per bandwidth, latency and load when intelligent is selected as the load balancing policy. By adjusting the weight base of bandwidth, latency and load, we can change the specific influence of these three factors on the link weight.

Configure the weight base of bandwidth, latency and load according to the following steps.

Command	Function
Ruijie# **configure terminal**	Enter global configuration mode.
Ruijie(config)# **mllb policy intelligent** [**bandwidth** *base1*] [**latency** *base2*] [**load** *base3*] }	Configure the weight bases of bandwidth, latency and load to base1, base2 and base3 respectively (1-100, with 1 being the default value).
Ruijie(config)# **no mllb policy intelligent**	Restore all weight bases to default value.

Configuration example:

# Configure load balancing policy to bandwidth, and configure the weight bases of bandwidth, latency and load to 20, 50 and 100 respectively:

```
Ruijie# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# mllb policy intelligent bandwidth 50 latency 20 load 100
```

Note

Weight base will only take effect when load balancing policy is configured to intelligent.

Displaying Configurations

In privilege mode, execute "show mllbconfig" command to display configurations related to multilink load balancing.

Command	Function
Ruijie (config)# **show mllb config**	Display system configurations, including load balancing configurations.

Configuration example:

# In privilege mode, execute "**show running-config**" command to display load balancing configurations:

```
Ruijie# show mllb config
muti-link load balance configure:
muti-link load balance state: enabled
```

```
muti-link load balance threshold: 95
muti-link load balance policy: intelligent
    bandwidth weight base = 100
    latency weight base = 100
    load weight base = 100
```

Typical Multilink Load Balancing Configuration Example

Networking Requirements

One 1000M telecom link is connected to interface gigabitEthernet 0/1 of the device, while two CNC links (with bandwidth being 100M and 10M respectively) are connected to interface gigabitEthernet 0/2 and interface gigabitEthernet 0/3 of the device.

Network Topology

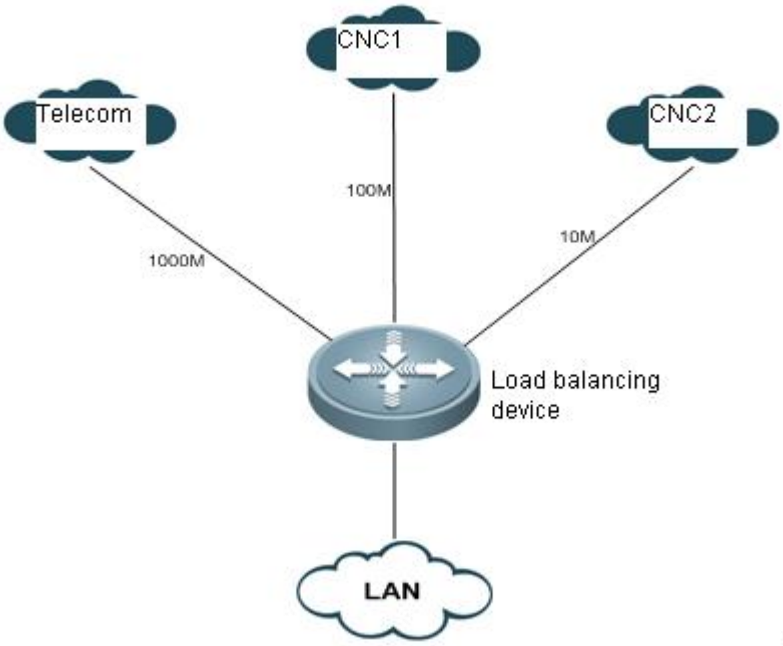


Fig 2 Networking topology of multilink load balancing

Configuration Steps

1) Enable multilink load balancing

# Enable global multilink load balancing

```
Ruijie# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# mllb enable
```

2) Configure link bandwidth

```
Ruijie# configure terminal
```

```
Enter configuration commands, one per line.  End with CNTL/Z.
```

# Configure the bandwidth of telecom link

```
Ruijie(config)# interface gigabitEthernet 0/1
Ruijie(config-if)# bandwidth 1000000
Ruijie# exit
```

# Configure the bandwidth of CNC link 1

```
Ruijie(config)# interface gigabitEthernet 0/2
Ruijie(config-if)# bandwidth 100000
Ruijie# exit
```

# Configure the bandwidth of CNC link 2

```
Ruijie(config)# interface gigabitEthernet 0/3
Ruijie(config-if)# bandwidth 10000
Ruijie# exit
```

3) Configure load balancing policy to bandwidth load balancing

```
# Configure load balancing policy to bandwidth
Ruijie(config)#mllb policy bandwidth
```

4) Configure link load threshold

# Configure link load threshold to 95

```
Ruijie(config)# mllb threshold 95
```

Configuring RNS & Track

Introduction to RNS

RNS (Ruijie Network Service) monitors the integrity of end-to-end connection by detecting whether the reply message is sent by the peer device. The icmp echo packets and dns request packets can be sent.

List of RNS Configuration Tasks

To configure rns, please follow these tasks:

- Configuring rns to detect icme echo
- Configuring rns to detect dns

Configuring RNS to Detect Icme Echo

General steps:

Configure a RNS object to send ICMP echo packets

1 **enable**
2 **configure terminal**
3 **ip rns** *operation-number*
4 **icmp-echo** destination-hostname [**source-ipaddr** *ip-address*] [**out-interface** *type number* [**next-hop** *nhop-ip*]]
5 **frequency** *milliseconds*
6 **timeout** *milliseconds*
7 **ntime** *number*
8 **exit**

Command	**Function**
Ruijie> **enable**	Enter privilege mode
Ruijie# **configure terminal**	Enter global configuration mode
Ruijie(config)# **ip rns** *operation-number*	Enter IP RNS configuration mode
Ruijie(config-ip-rns)# **icmp-echo destination-hostname** [*source-ipaddr* ip-address] [**out-interface** *type number* [**next-hop** *nhop-ip*]]	Configure an IP RNS object to send ICMP packets, and Enter icmp echo configuration mode If the optional parameter of “out-interface” is applied, note the following points: If out-interface is an Ethernet port, then next-hop has to be configured; if not, then next-hop is optional.
Ruijie(config-ip-rns-icmp-echo)**# frequency** *milliseconds*	(Optional) Set an interval of sending packets. The value of the interval should exceed or equal the timeout.

Ruijie(config-ip-rns-icmp-echo)# **timeout** *milliseconds*	(Optional) Set timeout of sending packets
Ruijie(config-ip-rns-icmp-echo)# **ntime** *number*	(Optional) Set detection times of packets

Configuration example:

```
Ruijie> enable
Ruijie# configure terminal
Ruijie(config)# ip rns 1
Ruijie(config-ip-rns)# icmp-echo 10.1.1.1
Ruijie(config-ip-rns-icmp-echo)# ntime 3
```

Display the configurations of RNS object:

```
Ruijie# show ip rns configuration
Ip rns id:1
Type of operation to perform: icmp-echo
Target address/Source address:10.1.1.1/0.0.0.0
Out interface/next hop: /0.0.0.0
Operation timeout (milliseconds):5000
Vrf Name:
Operation frequency (milliseconds):60000
Operation ntime: 3
```

Display statistics of RNS object:

```
Ruijie# show ip rns statistics
IP rns index   1
Number of successes:0
Number of failures:174
Round-trip min/avg/max = 0/0/0 ms
```

Configuring RNS to Detect DNS

General steps:

Configure a RNS object to send ICMP echo packets

1 **enable**
2 **configure terminal**
3 **ip rns** *operation-number*
4 **dns** *work* **name-server** *a.b.c.d*
5 **frequency** *milliseconds*
6 **timeout** *milliseconds*
7 **ntime** *number*
8 **exit**

Command	**Function**

Ruijie> **enable**	Enter privilege mode
Ruijie# **configure terminal**	Enter global configuration mode
Ruijie(config)# **ip rns** *operation-number*	Enter IP RNS configuration mode
Ruijie(config-ip-rns)# **dns** *work* **name-server** *a.b.c.d*	Configure an IP RNS object to send DNS packets and enter DNS configuration mode.
Ruijie(config-ip-rns-dns)# **frequency** *milliseconds*	(Optional) Set an interval of sending packets. The value of the interval should exceed or equal the timeout.
Ruijie(config-ip-rns-dns)# **timeout** *milliseconds*	(Optional) Set timeout of sending packets
Ruijie(config-ip-rns-dns)# **ntime** *number*	(Optional) Set detection times of packets

Configuration example:

```
Ruijie> enable
Ruijie# configure terminal
Ruijie(config)# ip rns 1
Ruijie(config-ip-rns) # dns www.ruijie.com name-server 1.1.1.1
Ruijie(config-ip-rns-icmp-echo) # frequency 50000
```

Display the configurations of RNS object:

```
Ruijie# show ip rns configuration
Ip rns id:1
Type of operation to perform: dns
Domain name: www.ruijie.com
Name server:1.1.1.1
Operation timeout (milliseconds):9000
Operation frequency (milliseconds):50000
Operation ntime: 1
```

Display statistics of RNS object:

```
Ruijie# show ip rns statistics
IP rns index   1
Number of successes:0
Number of failures:8
Round-trip min/avg/max = 0/0/0 ms
```

Introduction to Track

A track object can track whether an IP address is reachable and whether an interface is UP. The track feature separates a tracked object from the modules which are interested in this object, such as PBR and VRRP. When the states of track object changes, they can take different actions.

List of Track Configuration Tasks

To configure track, please follow these tasks:

- Tracking the link state of an interface
- Tracking the state of a RNS object

Tracking the Link State of an Interface

Execute this task to track the link state of an interface. A layer-2 interface is considered UP as long as the interface is powered up; a layer-3 interface is considered UP as long as the layer-2 interface of this interface-3 interface is UP; a logic interface like loopback interface is considered UP as long as it is not shut down.

General steps:

1. **enable**
2. **configure terminal**
3. **track** *object-number* **interface** *type number* **line-protocol**
4. **delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*}
5. **end**
6. **show track** *object-number*

Detailed steps:

Command	**Function**
Ruijie> **enable**	Enter privilege mode
Ruijie# **configure terminal**	Enter global configuration mode
Ruijie(config)# **track** *object-number* **interface** *type number* **line-protocol**	Track the state of an interface and enter track mode
Ruijie(config-track)# **delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*}	(Optional) Specify a delay time after which the state of track object will change when interface state changes. There is no delay by default.
Ruijie(config-track)# **show track** *object-number*	(Optional) Display the information about track object. You can use this command to verify whether the configurations are correct.

Configuration example:

```
Ruijie> enable
Ruijie# configure terminal
Ruijie(config)# track 3 interface FastEthernet 1/0 line-protocol
Ruijie(config-track)# delay up 30
Ruijie(config-track)# show track 3
```

Configure a track object to track the state of an interface. The following example shows relevant information:

```
Ruijie# show track 3
Track 3
```

```
interface FastEthernet 1/0
The state is Up
1 change,current state last:11 secs
Delay up 10 secs,down 10 secs
```

Tracking the State of a RNS Object

We uses a track object to track the state of RNS object; If the RNS object receives the reply packets, then the state of track object is UP; otherwise, the state of track object is DOWN.

Caution When track object is used to track a nonexistent RNS object, the state of this track object is UP.

General steps:

First configure an IP RNS object

1. **enable**
2. **configure terminal**
3. **ip rns** *operation-number*
4. **icmp-echo destination-hostname** [*source-ipaddr ip-address*]
5. **frequency** *seconds*
6. **exit**

Then configure a track object:

1. **enable**
2. **configure terminal**
3. **track** *object-number* **rns** *entry-number*
4. **delay up** *seconds* **down** *seconds*
5. **end**
6. **show track object-number**

Configure a route-map and apply the aforementioned track object:

1. **route-map map-tag [permit | deny]** [*sequence-number*]
2. **set ip next-hop verify-availability** [*next-hop-address sequence* **track** *object*]

Apply policy-based routing to an interface:

1. **interface** *type number*
2. **ip address** *ip-address* **mask** *[secondary]*
3. **ip policy route-map map-tag**
4. **exit**

Detailed steps:

Command	Function
Ruijie(config)# **track** *object-number* **rns** *entry-number*	Track the state of an IP RNS object and enter track mode.

Ruijie(config-track)# **delay** {**up** *seconds* [**down** *seconds*] | [**up** *seconds*] **down** *seconds*}	(Optional) Specify a delay time after which the state of track object will change when its state changes. There is no delay by default.
Ruijie(config-track)# **exit**	Return to global configuration mode.
Ruijie(config)# **interface** *type number*	Enter interface configuration mode.
Ruijie(config-if)# **ip address** *ip-address* **mask** [*secondary*]	Configure an IP address for the interface.
Ruijie(config-if)# **ip policy route-map map-tag**	Apply policy-based routing to this interface.
Ruijie(config-if)# **exit**	Return to global configuration mode.
Ruijie(config)# **route-map map-tag** [**permit | deny**] [*sequence-number*]	Configure a route-map.
Ruijie(config-route-map)# **set ip next-hop verify-availability** [*next-hop-address sequence* **track** *object*]	The route map configured is used to track the state of a track object

Configuration example:

```
Ruijie(config)# track 123 rns 1
Ruijie(config-track)# delay up 30
Ruijie(config-track)# exit
Ruijie(config)# interface ethernet 0
Ruijie(config-if)# ip address 10.1.1.11 255.0.0.0
Ruijie(config-if)# ip policy route-map alpha
Ruijie(config-if)# exit
Ruijie(config)# route-map alpha
Ruijie(config-route-map)# set ip next-hop verify-availability 10.1.1.1 10 track 123
```

Display the state of a track object:

```
Track 2
Ruijie Network Service 1
The state is Down
1 change,current state last:7 secs
Delay up 30 secs,down 0 secs
```

Application of Track Feature

We can achieve the following function by tracking the UP/DOWN state of track object:

■ Associating the next hop of PBR with a track object

Associating the Next Hop of PBR with a Track Object

In policy-based routing, we can associate the next hop of PBR with a track object. When the state of track object becomes DOWN, this next hop will be disabled, namely PBR will not use this next hop as the next hop for packets.

General steps:

First configure an IP RNS object (please refer RNS configurations given above).

Then configure a track object (please refer to track configurations given above).

Configure a route-map and apply the track object configured above:

1. **route-map map-tag [permit | deny]** [*sequence-number*]
2. **set ip next-hop verify-availability track** *object*

Apply policy-based routing to an interface:

1. **interface** *type number*
2. **ip address** ip-address **mask** [secondary]
3. **ip policy route-map map-tag**
4. **exit**

Detailed steps:

Command	**Function**
Ruijie(config)# **interface** *type number*	Enter interface configuration mode.
Ruijie(config-if)# **ip address** *ip-address mask* [*secondary*]	Configure an IP address for the interface.
Ruijie(config-if)# **ip policy route-map** *map-tag*	Apply policy-based routing to this interface.
Ruijie(config-if)# **exit**	Return to global configuration mode.
Ruijie(config)# **route-map** *map-tag* [**permit** | **deny**]	Configure a route-map.
Ruijie(config-route-map)# **set ip next-hop verify-availability track** *object*	The route map configured is used to track the state of a track object.

Configuration example:

```
Ruijie(config)# interface ethernet 0
Ruijie(config-if)# ip address 10.1.1.11 255.0.0.0
Ruijie(config-if)# ip policy route-map alpha
Ruijie(config-if)# exit
Ruijie(config)# route-map alpha
Ruijie(config-route-map)# set ip next-hop verify-availability 10.1.1.1  track 123
```

The policy-based routing to be achieved: When packets are received by fa 0/0 and the IP address of 10.1.1.1 is reachable, the next hop of packets is configured as 10.1.1.1 (IP address of the interface on router 2). If 10.1.1.1 is unreachable, the next hop of packets is configured as 10.2.2.2 (IP address of the interface on router 3). If 10.2.2.2 is also unreachable, PBR fails. Packets will be forwarded as per ordinary route according to the query result of core routing table.

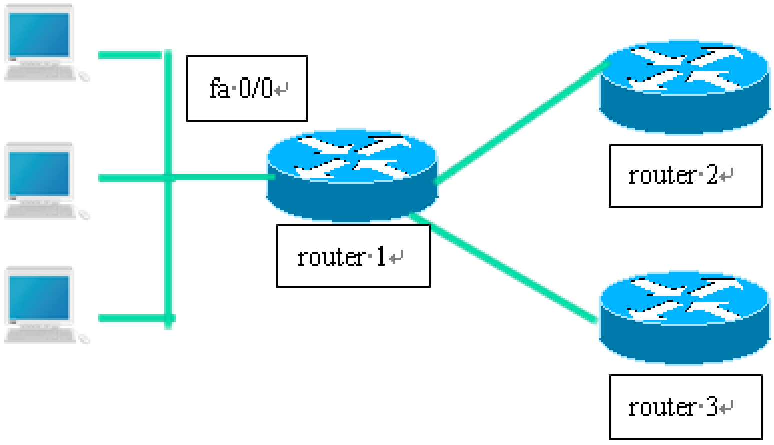


Figure 1

Configurations of Router 1:

# Define two IP RNS objects to track whether the remote IP address is reachable.

```
ip rns 1
icmp-echo 10.1.1.1
ip rns 2
icmp-echo 10.2.2.2
```

# Define track object

```
track 123 rns 1
track 124 rns 2
```

# Apply PBR to the interface

```
interface FastEthernet 0/0
ip address 10.4.4.4 255.255.255.0
ip policy route-map alpha
```

# 10.1.1.1 is the following interface

```
interface fa 0/1
ip address 10.1.1.254 255.255.255.0
```

# 10.2.2.2 is the following interface

```
interface fa 0/2
ip address 10.2.2.254 255.255.255.0
```

# Configure a route-map; the availability of next hop depends on the reachability of track object.

```
route-map alpha
set ip next-hop verify-availability 10.1.1.1  track 123
set ip next-hop verify-availability 10.2.2.2  track 124
```

IPv6 Configuration

Configuring IPv6

Understanding IPv6

Overview

As the Internet is growing rapidly and the IPv4 address space is exhausting, the limitation of the IPv4 is more obvious. The research and practice of the next generation Internet Protocol (IPng) become popular. Furthermore, the IPng working group of the IETF has determined the protocol specification of IPng referred to as IPv6. See RFC 2460 for details.

Key Features

- More address space

The length of an address is extended to 128 bits from 32 bits of IPv4. Namely, there are 2^128-1 addresses for IPv6. IPv6 adopts hierarchical address mode and supports multiple-level IP address assignment, for example, from the Internet backbone network to the internal subnet of enterprises.

- Simplified format of packet header

The design principle of the new IPv6 packet header is to minimize the overhead. For this reason, some non-critical fields and optional fields are removed from the packet header and placed into the extended packet header. The length of an IPv6 address is 4 times the length of an IPv4 address; the size of the IPv6 packet header is only 2 times the size of the IPv4 packet header. The improved IPv6 packet header is more efficient for forwarding, for example, there is no checksum in the IPv6 packet header and it is not necessary for an IPv6 device to process the fragments during forwarding (the fragments are completed by the originator).

- High-efficient hierarchical addressing and routing structure

IPv6 adopts the aggregation mechanism and defines a flexible hierarchical addressing and routing structure, and several networks at the same level are represented with a unified network prefix at a higher-layer device. So it obviously reduces the routing entries that the device must maintain and greatly minimizes the routing and storage overhead.

- Simple management: plug and play

The management and maintenance of network nodes are simplified by the implementation of a series of auto-discovery and auto-configuration functions. For example, the neighbor discovery, MTU discovery, router advertisement (RA), router solicitation (RS) and auto-configuration technologies provide the related service for plug and play. It should be mentioned that IPv6 supports such address configuration methods as stateful configuration and stateless configuration. In IPv4, the Dynamical Host Configuration Protocol (DHCP) implements the automatic configuration of a host IP address and related configuration, while IPv6 inherits this auto-configuration service of IPv4 and refers to it as the stateful auto-configuration. Furthermore, IPv6 also adopts an auto-configuration service, referred to as stateless auto-configuration. During the stateless auto-configuration, the host obtains the link-local address, the address prefix of the local device and some other related configuration information automatically.

- Security

IPSec is an optional extended protocol of IPv4, but it is only a component of IPv6 used to provide security. At present, IPv6 implements the authentication header (AH) and encapsulated security payload (ESP) mechanisms. The former authenticates the integrity of data and the source of an IP packet to ensure that the packet does come from the node marked by the source address, while the latter provides the data encryption function to implement end-to-end encryption.

- More excellent QoS support

A new field in the IPv6 packet header defines how to identify and process a data flow. The Flow Label field in an IPv6 packet header is used to identify the data flow ID, by which IPv6 allows users to put forward the requirement for the QoS of communication. The device can identify all packets of a specified data flow by this field and provide special processing for these packets as required.

- New protocol for interactions between neighbor nodes

The Neighbor Discovery Protocol of IPv6 uses a series of IPv6 control information messages (ICMPv6) to manage the interactions between neighbor nodes (the nodes on the same link). The Neighbor Discovery Protocol and high-efficient multicast and unicast neighbor discovery messages replace the previous broadcast-based Address Resolution Protocol (ARP) and the ICMPv4 router discovery messages.

- Extensibility

IPv6 provides powerful extensibility and the new features can be added to the extended packet header after the IPv6 packet header. Unlike the IPv4 packet header, the IPv6 packet header can only support the options of up to 40 bytes, while the size of the IPv6 extended packet header is only limited by the maximum number of bytes of the whole IPv6 packet.

IPv6 supports the following features:

- IPv6 protocol
- IPv6 address format
- Type of IPv6 address
- ICMPv6
- IPv6 neighbor discovery
- Path MTU discovery
- ICMPv6 redirection
- Duplicate address detection
- IPv6 stateless auto-configuration
- IPv6 address configuration
- IPv6 route forwarding (supporting static route configuration)
- Configuration of various IPv6 parameters
- Diagnosis tool ping IPv6

IPv6 Address Format

The basic format of an IPv6 address is X : X : X : X : X : X : X : X, where X is a 4-digit hexadecimal integer (16 bits). Each digit contains 4 bits, each integer contains 4 hexadecimal digits, and each address contains 8 integers, so the address includes a total of 128 bits. Some legal IPv6 addresses are as follows:

2001:ABCD:1234:5678:AAAA:BBBB:1200:2100

800 : 0 : 0 :0 : 0 : 0 : 0 : 1

1080 : 0 : 0 : 0 : 8 : 800 : 200C : 417A

These integers are hexadecimal integers, where A to F denote 10 to 15 respectively. Each integer in the address must be denoted and the starting 0 need not be denoted. Some IPv6 addresses may contain a series of 0s (such as the second and third examples). In this case, colons (: :) are allowed to denote this series of 0s. Namely, the address 800:0:0:0:0:0:0:1 can be denoted as: 800 :: 1.

These two colons denote that this address can be extended to a complete 128-bit address. In this way, the 16-bit group can be replaced with two colons only when they are all 0s and the two colons can only be present once.

In the hybrid environment of IPv4 and IPv6, there is a hybrid denotation method. The lowest 32 bits in an IPv6 address can be used to denote an IPv4 address. The address can be expressed in a hybrid mode, that is, X: X : X : X : X : X : d . d . d . d, where, X denotes a 16-bit integer, while d denotes an 8-bit decimal integer. For instance, the address 0 : 0 : 0 : 0 : 0 : 0 : 192 .168 . 20 : 1 is a legal IPv6 address. After the abbreviated expression method is used, this address can be denoted as follows: : : 192.168. 20. 1. One typical example is an IPv4-compatible IPv6 address, which is expressed as "::A.B.C.D", with the first 96 bits being all 0s, such as "::1.1.1.1", but this expression method is revoked. Another typical example is an IPv4-mapped IPv6 address, which is expressed as "::FFFF:A.B.C.D" and used to express an IPv4 address as an IPv6 address, that is, map the IPv4 address"1.1.1.1" to the IPv6 address"::FFFF:1.1.1.1".

Because the IPv6 address is divided into two parts, the subnet prefix and the interface identifier, it can be denoted as an address including an additional numeric value by the method like the CIDR address. This numeric value indicates how many bits represent the network part (the network prefix). Namely the IPv6 node address indicates the length of the prefix, and the length is differentiated from the IPv6 address by a slash. For instance: 12AB::CD30:0:0:0:0/60. The length of the prefix used for routing in this address is 60 bits.

Type of IPv6 Address

RFC 4291 defines three types of IPv6 addresses:

- Unicast: Identifier of a single interface. The packet to be sent to a unicast address will be transmitted to the interface identified by this address.
- Anycast: Identifiers of a set of interfaces. The packet to be sent to an anycast address will be transmitted to one of the interfaces identified by this address (the nearest one is selected according to the routing protocol).
- Multicast: Identifiers of a set of interfaces (In general, these interfaces belong to different nodes). The packet to be sent to a multicast address will be transmitted to all the interfaces that join this multicast address.

Caution The broadcast address is not defined in IPv6.

The following describes these types of addresses one by one.

Unicast Addresses

The unicast addresses are divided into unspecified address, loopback address, link-local address, site-local address and global unicast address. Now the site-local address has been revoked. The unicast addresses excepting the unspecified address, loopback address and link-local address are all global unicast addresses.

1) Unspecified address

The unspecified address is 0:0:0:0:0:0:0:0, generally abbreviated as :: and used for the following purposes.

- If there is no unicast address when a host is started, use the unspecified address as the source address, send an RS, and obtain the prefix information from the gateway to automatically generate the unicast address.
- When configuring an IPv6 address for the host, check whether the IPv6 address conflicts with the address of any other host in the same network segment or not. If so, use the unspecified address as the source address to send a neighbor solicitation (NS) message, same as free ARP.

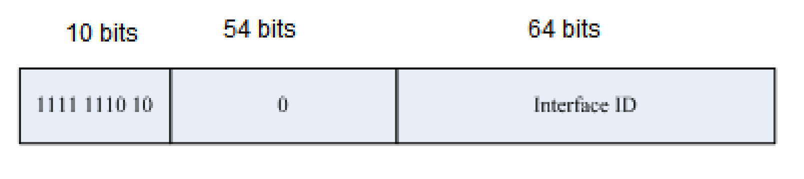
2) Loopback address

The loopback address is 0:0:0:0:0:0:0:1, abbreviated as ::1, which is equal to the IPv4 address 127.0.0.1 and used when the node sends the packets to itself.

3) Link–local address

The format of link-local address:

Figure 1

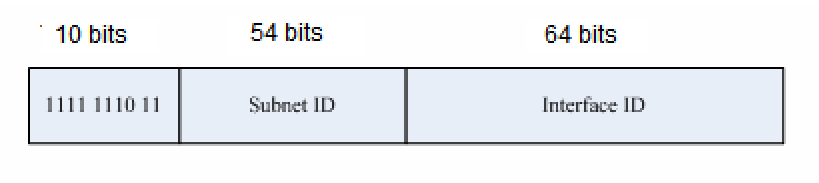


The link-local address is used to number the host on the single network link. The address identified by the first 10 bits of the prefix is the link-local address. The device will never forward the packet of the source address or the destination address with the link-local address. The intermediate 54 bits are all 0s. The last 64 bits indicate the interface identifier, and this part allows the single network to connect up to 2^{64}-1 hosts.

4) Site-local address

The format of site-local address:

Figure 2

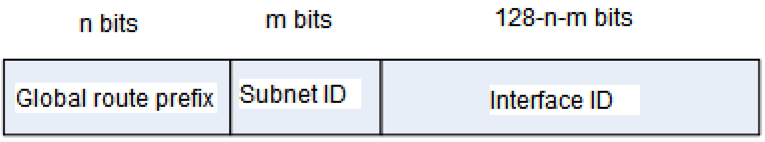


The site-local address can be used to transmit data within the site, and the device will not forward the packet of the source address or the destination address with the site-local address to the Internet. Namely, such packet can only be forwarded within the site, but cannot be forwarded out of the site. The site may be deemed as the LAN of a company, and the site-local address is similar to a private IPv4 address, for example, 192.168.0.0/16. RFC 3879 has revoked the site-local address. In new implementations, this prefix is no longer supported and is uniformly deemed as a global unicast address. In existing implementations and deployments, this prefix may be still used.

5) Global unicast address

The format of global unicast address:

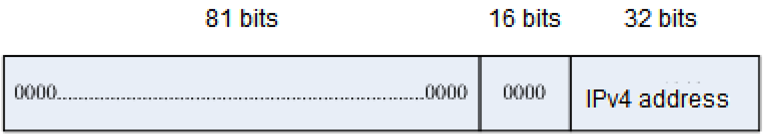
Figure 3



One class of the global unicast address is the IPv6 address embedded with an IPv4 address, which is used to interconnect the IPv4 nodes and the IPv6 nodes and divided into IPv4-compatible IPv6 address and IPv4-mapped IPv6 address.

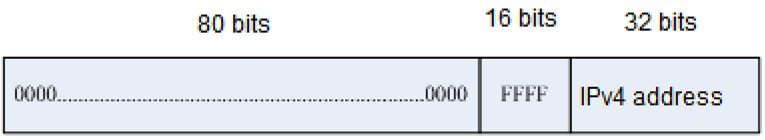
The format of IPv4-compatible IPv6 address:

Figure 4



The format of IPv4-mapped IPv6 address:

Figure 5



The IPv4-compatible IPv6 address is mainly used for automatic tunneling, which supports both IPv4 and IPv6. The IPv4-compatible IPv6 address is used to transmit an IPv6 packet via an IPv4 device in the tunneling way. Now the IPv4-compatible IPv6 address has been revoked. The IPv4-mapped IPv6 address is used by IPv6 nodes to access the nodes that only support IPv4. For example, when one IPv6 application of the IPv4/IPv6 host requests the resolution of a host name (the host only supports IPv4), the name server will internally generate an IPv4-mapped IPv6 address dynamically and return it to the IPv6 application.

Multicast Addresses

The format of the IPv6 multicast address is as follows:

```
|    8     |  4 |  4 |                      112 bits                       |

+----------+----+----+-------------------------------------------------------------+

|11111111|flgs|scop|                         group ID                       |

+----------+----+----+-------------------------------------------------------------+
```

The first byte of the address format is all 1s, which denote a multicast address.

- Flag field:

It consists of 4 bits. At present, only the fourth bit is specified. The bit is used to indicate whether the address is a known multicast address specified by the Internet Assigned Numbers Authority (IANA) or a temporary multicast address used on a specific occasion. If this flag bit is 0, it indicates this address is a known multicast address. If this bit is 1, it indicates that this address is a temporary one. Other 3 flag bits are reserved for future use.

- Range field:

The Range field is composed of 4 bits and used to denote the range of multicast, namely, whether the multicast group contains the local node, the local link and the local site or nodes in any positions in the IPv6 global address space.

- Group ID field:

This field is 112 bits long and used to identify a multicast group. Depending on whether a multicast address is temporary or known and the range of the address, a multicast identifier can denote different groups.

The multicast address of IPv6 is this type of address using FF00::/8 as the prefix. One multicast address of IPv6 usually identifies the interfaces of a serial of different nodes. When one packet is sent to one multicast address, this packet will be distributed to the interfaces of each node with this multicast address. One node (host or device) should join the following multicast addresses:

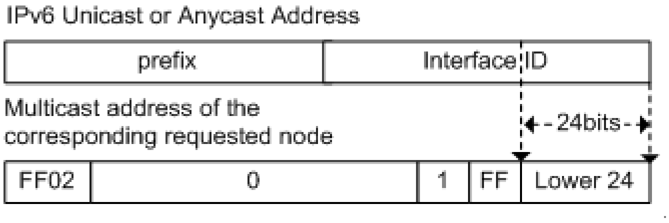
- The multicast address of all nodes on the local link, that is, FF02::1
- The multicast address of the solicited node, with the prefix of FF02:0:0:0:0:1:FF00:0000/104

For the device, it is necessary to join the multicast address FF02::2 of all devices on the local link.

If the multicast address of the solicited node corresponds to the IPv6 unicast and anycast address, it is necessary for the IPv6 node to join the corresponding multicast address of the solicited node for each configured unicast address and anycast address. The prefix of the multicast address of the solicited node is FF02:0:0:0:0:1:FF00:0000/104, another 24 bits are comprised of the unicast address or the lower 24 bits of the anycast address, for example, the multicast address of the solicited node corresponding to the unicast address FE80::2AA:FF:FE21:1234 is FF02::1:FF21:1234.

The multicast address of the solicited node is usually used in an NS message. The format of the solicited node is as follows:

Figure 6


IPv6 Unicast or Anycast Address
prefix
Interface ID
Multicast address of the corresponding requested node
24bits
FF02
0
1
FF
Lower 24

Anycast Addresses

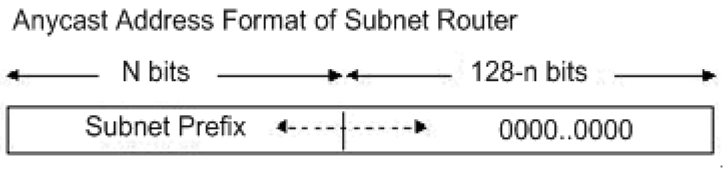
The anycast address is similar to the multicast address as more than one node shares an anycast address. The difference is that only one node expects to receive the data packet of the anycast address, while all nodes of the multicast group members expect to receive all packets sent to this address. The anycast address is assigned to the normal IPv6 unicast address space, so the anycast address cannot be differentiated from the unicast address from the style. For this reason, each member of an anycast group represented by an anycast address must be configured explicitly to identify the anycast address.

Caution The anycast address can only be assigned to the device, but cannot be assigned to the host. Furthermore, the anycast address cannot be used as the source address of the packet.

RFC 2373 predefines an anycast address, referred to as the anycast address of a subnet router. The following figure shows the anycast address format of the subnet router, which consists of the subnet prefix followed by a series of 0s (as the interface identifier).

The subnet prefix identifies a specified link (subnet) and the packet to be sent to the anycast address of the subnet router will be distributed to a device of this subnet. The anycast address of the subnet router is usually used for a node which needs to communicate with one device of a remote subnet.

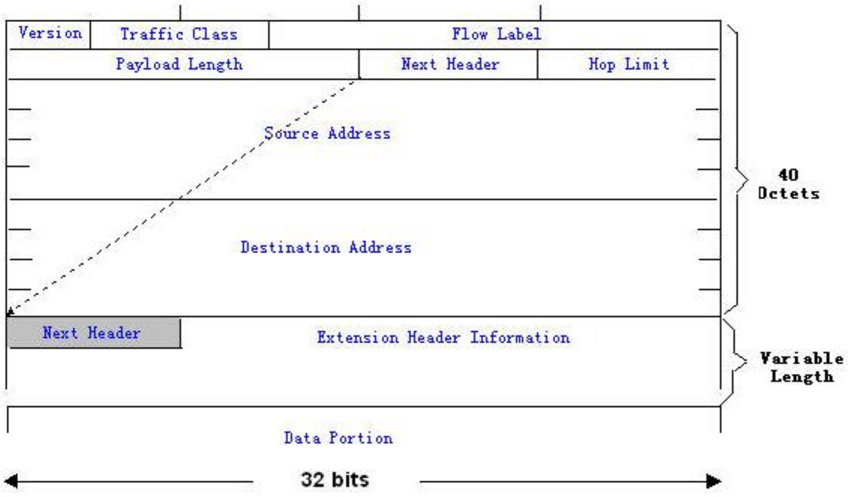
Figure 7



IPv6 Packet Header Structure

The format of the IPv6 packet header is shown in the following figure.

Figure 8



In IPv4, the packet header is measured in units of 4 bytes; in IPv6, the packet header is measured in units of 8 bytes, and the total size of the packet header is 40 bytes. In the IPv6 packet header, the following fields are defined:

- Version:

The length is 4 bits. For IPv6, the field must be 6.

- Traffic Class:

The length is 8 bits. It indicates a type of service provided to the packet and is equal to the “TOS” in IPv4.

- Flow Label:

The length is 20 bits. This field is used to identify the packets of the same service flow. One node can be used as the source of several service flows. The flow label and source node IP address identify a service flow uniquely.

- Payload Length:

The length is 16 bits, including the byte length of the payload and the length of various IPv6 extension options (if any). In other words, it includes the length of an IPv6 packet except for the IPv6 header.

- Next Header:

This field indicates the protocol type in the header field following the IPv6 header. Similar to the IPv4 protocol field, the Next Header field can be used to indicate whether the upper layer protocol is TCP or UDP. It can also be used to indicate whether an extended IPv6 header exists.

- Hop Limit:

The length is 8 bits. When the device forwards the packet for one time, the value of this field will decrease by 1. When the value of this field is 0, this packet will be discarded. It is similar to the lifetime field in the IPv4 packet header.

- Source Address:

The length is 128 bits. It indicates the sender address of an IPv6 packet.

- Destination Address:

The length is 128 bits. It indicates the receiver address of an IPv6 packet.

At present, the following extended headers are defined in IPv6:

- Hop-by-Hop Options:

This extended header must immediately follow an IPv6 header. It contains the option data that must be checked by each node along the path.

- Routing Header (Routing (Type 0)):

This extended header indicates the nodes that a packet will go through before reaching the destination. It contains the addresses of various nodes that the packet goes through. The initial destination address of the IPv6 header is the first one of a series of addresses in the routing header, other than the final destination address of the packet. After receiving this packet, the node of this address will process the IPv6 header and the routing header, and send the packet to the second address in the routing header. This process continues until the packet reaches the final destination.

- Fragment:

This extended header is used to fragment the packets longer than the MTU of the path between the source node and destination node.

- Destination Options:

This extended header replaces the IPv4 option field. At present, the only defined destination option is an option to be filled with an integral multiple of 64 bits (8 bytes) when necessary. This extended header can be used to carry the information checked by the destination node.

- Upper-layer header:

It indicates the upper layer transmission protocol, such as TCP(6) and UDP(17).

Furthermore, the extended header of Authentication and Encapsulating Security Payload will be described in the IPSec section. At present, the IPv6 implemented by the device does not support IPSec.

IPv6 Path MTU Discovery

Similar to the path MTU discovery of IPv4, the path MTU discovery of IPv6 allows one host to discover and adjust the size of the MTU in the data transmission path.

Furthermore, when the data packet to be sent is larger than the MTU of the data transmission path, the host will fragment the packet by itself. This behavior makes it not necessary for the device to process the fragment, and thus save resources and improve the efficiency of the IPv6 network.

Caution

The minimum link MTU is 68 bytes in IPv4, indicating that the links along the path over which the packets are transmitted should support at least the link MTU of 68 bytes. The minimum link MTU is 1280 bytes in IPv6. It is strongly recommended that the link MTU of 1500 bytes should be used for the link in IPv6.

IPv6 Neighbor Discovery

The main functions of the IPv6 Neighbor Discovery Protocol include router discovery, prefix discovery, parameter discovery, address auto-configuration, address resolution (ARP), next-hop determination, neighbor unreachability detection, duplicate address detection, and redirection. Neighbor discovery defines 5 types of ICMP messages, which are router solicitation (ICMP type133), RA (ICMP type134), NS or ARP request (ICMP type135), neighbor advertisement or APR response (ICMP type136) and ICMP redirection message (ICMP type137).

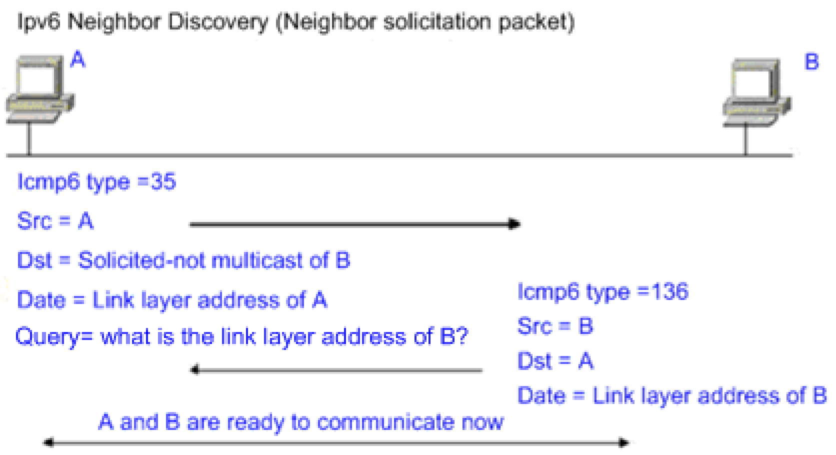
The following describes the neighbor discovery function in detail:

Address Resolution

A node must obtain the link layer address of another node before communicating with it. At this time, the node should send an NS message to the solicited multicast address, that is, the IPv6 address of the destination node. The NS message also contains the link layer address of itself. After receiving this NS message, the destination node responds with a message, referred to as neighbor advertisement (NA), with its link layer address. After receiving the response message, the source node can communicate with the destination node.

The following is the address resolution procedure:

Figure 9


Ipv6 Neighbor Discovery (Neighbor solicitation packet)
A
B
Icmp6 type =35
Src = A
Dst = Solicited-not multicast of B
Date = Link layer address of A
Query= what is the link layer address of B?
Icmp6 type =136
Src = B
Dst = A
Date = Link layer address of B
A and B are ready to communicate now

Neighbor Unreachability Detection

When the reachable time of a neighbor expires, neighbor unreachability detection is performed if an IPv6 unicast packet needs to be sent to this neighbor.

Neighbor unreachability detection and sending the IPv6 packet to the neighbor can be performed concurrently. During the detection, the device continues to forward the IPv6 packet to the neighbor.

Duplicate Address Detection

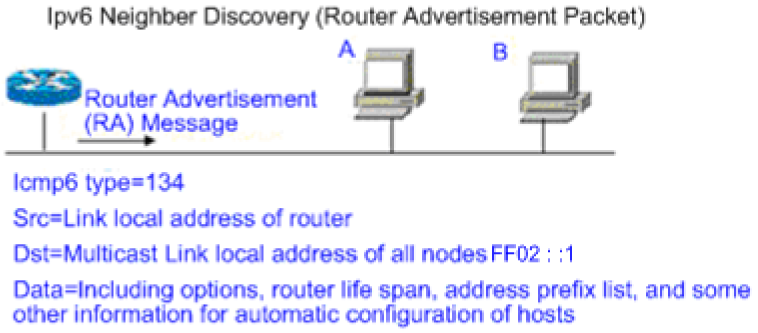
After an IPv6 address is configured for the host, duplicate address detection may be performed to know whether the IPv6 address is unique on the link, by sending an NS message with the source IPv6 address being an unspecified address.

Router, Prefix and Parameter Discovery

The router sends an RA to all the local nodes of the link periodically.

The following figure shows the process of sending the RA.

Figure 10


Ipv6 Neighbor Discovery (Router Advertisement Packet)
A
B
Router Advertisement
(RA) Message
Icmp6 type=134
Src=Link local address of router
Dst=Multicast Link local address of all nodes FF02 : :1
Data=Including options, router life span, address prefix list, and some other information for automatic configuration of hosts

In general, the RA contains the following contents:

- One or more IPv6 address prefixes used for the on-link determination or the stateless address auto-configuration.
- Effective period of the IPv6 address prefix.
- Host auto-configuration mode (stateful or stateless).

- Information for the default device (namely, the device determines whether it is used as the default device. If yes, it will announce the time to act as the default device).
- Other information for host configuration such as the hop limit, the MTU and the NS retransmission interval.

The RA is also used to respond to the RS message sent by the host. The RS message allows the host to obtain the auto-configuration information immediately without waiting for the device to send the RA. If there is no unicast address when the host is started, the RS message sent by the host will use the unspecified address (0:0:0:0:0:0:0:0) as the source address of the RS message. Otherwise, the existing unicast address is used as the source address, while the RS message uses the multicast address (FF02::2) of all devices on the local link as the destination address. The RA message, in response to the RS message, will use the source address of the RS message as the destination address (if the source address is the unspecified address, it will use the multicast address FF02::1) of all nodes on the local link.

The following parameters can be configured in the RA message.

Ra-interval: interval for sending the RA

Ra-lifetime: router lifetime, namely, whether the device acts as the default router of the local link and the time to act this role

Prefix: IPv6 address prefix of the local link, which can be used for the on-link determination or the stateless address auto-configuration, including the configuration of other parameters for the prefix

Rs-initerval: interval for sending the NS message

Reachabletime: time maintained after the neighbor is considered reachable

The above parameters are configured in the IPv6 interface properties.

⚠ Caution

1. No RA message is sent actively on the interface by default. To allow the device to send the RA message, you can use the **no ipv6 nd suppress-ra** command in interface configuration mode.
2. In order to enable normal stateless address auto-configuration of the node, the length of the prefix for the RA message should be 64 bits.

Redirection

After receiving the IPv6 packets, the router discovers an optimal next hop and sends an ICMP redirection message to notify the host of the optimal next hop. Next time the host sends the IPv6 packets to the optimal next hop directly.

Configuring IPv6

The following will describe the configuration of various functional modules of IPv6 respectively.

Configuring an IPv6 Address

This section describes how to configure an IPv6 address on an interface. No IPv6 address is configured by default.

Caution

Once an interface is created and its link state is UP, the system will automatically generate the link-local address for the interface. At present, IPv6 does not support anycast address.

For S57 and S76 series, the range of the length of the prefix of the interface IPv6 address is [0, 64] or [128, 128], because the range of the length of the routing prefix supported by the hardware forwarding table of the chip is [0, 64] or [128, 128]. For S86 and S96 series, the range of the length of the prefix of the interface IPv6 address is not limited, but the total number of IPv6 routes within the range [65, 127] of the length of the routing prefix supported by switches is 512.

To configure an IPv6 address, use the following commands.

Command	Function
Ruijie#**configure terminal**	Enters global configuration mode.
Ruijie(config)#**interface** *interface-id*	Enters interface configuration mode. Note that the **no switchport** command must be used to switch the layer-2 interface to the layer-3 interface.
Ruijie(config-if)#**ipv6 enable**	Enables the IPv6 protocol on an interface. If this command is not run, the system automatically enables the IPv6 protocol when you configure an IPv6 address for an interface.
Ruijie(config-if)#**ipv6 address** *ipv6-address/prefix-length* Ruijie(config-if)#**ipv6 address** *ipv6-prefix/prefix-length* [**eui-64**]	Configures an IPv6 unicast address for this interface. When the command includes the keyword **Eui-64**, only the prefix must be specified, and the interface ID is automatically generated in the EUI-64 format. The generated IPv6 address consists of the configured address prefix and the 64-bit interface ID. Note: Whether the keyword **eui-64** is used, it is necessary to enter the complete address format to delete an IPv6 address (prefix + interface ID or prefix length). When you configure an IPv6 address on an interface, the IPv6 protocol is automatically enabled on the interface. Even if you use **no ipv6 enable**, you cannot disable the IPv6 protocol.
Ruijie(config-if)#**end**	Returns to privileged EXEC mode.
Ruijie#show **ipv6 interface** *interface-id*	Displays the IPv6 interface information.
Ruijie#copy **running-config startup-config**	Saves the configuration.

To delete the configured IPv6 address, use the **no ipv6 address** *ipv6-prefix*I*prefix-length* **[eui-64]** command.

The following example configures an IPv6 address.

```
Ruijie(config)# interface GigabitEthernet 0/1
Ruijie(config-if)# ipv6 enable
Ruijie(config-if)# ipv6 address fec0:0:0:1::1/64
Ruijie(config-if)# end
Ruijie(config-if)# show ipv6 interface GigabitEthernet 0/1
Interface GigabitEthernet 0/1 is Up, ifindex: 1
address(es):
Mac Address: 00:00:00:00:00:01
INET6: fe80::200:ff:fe00:1 , subnet is fe80::/64
```

```
INET6: fec0:0:0:1::1 , subnet is fec0:0:0:1::/64
Joined group address(es):
ff01:1::1
ff02:1::1
ff02:1::2
ff02:1::1:ff00:1
MTU is 1500 bytes
ICMP error messages limited to one every 100 milliseconds
ICMP redirects are enabled
ND DAD is enabled, number of DAD attempts: 1
ND reachable time is 30000 milliseconds
ND advertised reachable time is 0 milliseconds
ND retransmit interval is 1000 milliseconds
ND advertised retransmit interval is 0 milliseconds
ND router advertisements are sent every 200 seconds<160--240>
ND router advertisements live for 1800 seconds
```

Configuring ICMPv6 Redirection

This section describes how to configure the ICMPv6 redirection function on the interface. The redirection function of the IPv6 on the interface is enabled by default. The device needs to send a redirection message to the initiator during packet forwarding in the following cases:

- The destination address of the message is not a multicast address.
- The destination address of the message is not the device itself.
- The output interface of the next hop determined by the device for this message is the same as the interface that receives this message, namely, the next hop and the initiator are on the same link.
- The node identified by the source IP address of the packet is a neighbor of the local device. Namely, this node exists in the device’s neighbor table.

Caution

The device other than the host can generate the redirection message, and the device will not update its routing table when it receives the redirection message.

To enable redirection on the interface, use the following commands.

Command	Function
Ruijie#**configure terminal**	Enters global configuration mode.
Ruijie(config)#**interface** *interface-id*	Enters interface configuration mode. Note that the **no switchport** command must be used to switch the layer-2 interface to the layer-3 interface.
Ruijie(config-if)#**ipv6 redirects**	Enables the IPv6 redirection function.
Ruijie(config-if)#**end**	Returns to privileged EXEC mode.
Ruijie#**show ipv6 interface** *interface-id*	Displays the interface configuration information.
Ruijie#**copy running-config startup-config**	Saves the configuration.

To disable the redirection function, use the **no ipv6 redirects** command.

The example configures the redirection function.

```
Ruijie(config)# interface GigabitEthernet 0/1
Ruijie (config-if)# ipv6 redirects
Ruijie (config-if)# end
Ruijie # show ipv6 interface GigabitEthernet 0/1
Interface GigabitEthernet 0/1 is Up, ifindex: 1
address(es):
Mac Address: 00:d0:f8:00:00:01
INET6: fe80::2d0:f8ff:fe00:1 , subnet is fe80::/64
INET6: fec0:0:0:1::1 , subnet is fec0:0:0:1::/64
Joined group address(es):
ff01:1::1
ff02:1::1
ff02:1::2
ff02:1::1:ff00:1
MTU is 1500 bytes
ICMP error messages limited to one every 10 milliseconds
ICMP redirects are enabled
ND DAD is enabled, number of DAD attempts: 1
ND reachable time is 30000 milliseconds
ND advertised reachable time is 0 milliseconds
ND retransmit interval is 1000 milliseconds
ND advertised retransmit interval is 0 milliseconds
ND router advertisements are sent every 200 seconds<160--240>
ND router advertisements live for 1800 seconds
```

Configuring a Static Neighbor

This section describes how to configure a static neighbor. No static neighbor is configured by default. In general, a neighbor learns and maintains its status by the Neighbor Discovery Protocol (NDP) dynamically. Moreover, you can configure the static neighbor manually.

To configure the static neighbor, use the following commands.

Command	Function
Ruijie#**configure terminal**	Enters global configuration mode.
Ruijie(config)#**ipv6 neighbor** *ipv6-address interface-id hardware-address*	Configure a static neighbor on the interface.
Ruijie(config)#**end**	Returns to privileged EXEC mode.
Ruijie#**show ipv6 neighbors**	View the neighbor list.
Ruijie#**copy running-config startup-config**	Saves the configuration.

To delete the specified neighbor, use the **no ipv6 neighbor** *ipv6-address interface-id* command.

The following example configures a static neighbor on the GigabitEthernet 0/1 interface.

```
Ruijie(config)# ipv6 neighbor fec0:0:0:1::100 GigabitEthernet 0/1 00d0.f811.1234
Ruijie (config)# end
```

```
Ruijie# show ipv6 neighbors verbose fec0:0:0:1::100
IPv6 Address       Linklayer Addr   Interface
fec0:0:0:1::100     00d0.f811.1234   GigabitEthernet 0/1
State: REACH/H Age: - asked: 0
```

Caution

When you configure a static neighbor, the configuration takes effect only when the neighbor prefix matches the interface. Specifically, the configured static neighbor prefix belongs to the network segment of an address configured for the interface, and does not conflict with the address. An invalid static neighbor is in the inactive state. Data sent to the destination is not sent to the MAC address specified by the static neighbor, but the MAC address is learned based on routes in dynamic learning mode. To view the validity status of a static neighbor, run the **show ipv6 neighbor static** command.

Configuring Duplicate Address Detection

This section describes how to configure duplicate address detection times. Duplicate address detection is mandatory to assign unicast addresses to interfaces. The purpose is to detect the uniqueness of an address. Duplicate address detection should be performed for addresses that are configured in manual configuration mode, stateless auto-configuration mode, and stateful auto-configuration mode. However, it is not necessary to perform duplicate address detection under the following two conditions:

- The management prohibits the duplicate address detection, namely, the number of the NS messages sent for the duplicate address detection is set to 0.
- Duplicate address detection cannot be performed for a configured anycast address.

Furthermore, if the duplicate address detection function is not disabled on the interface, the system will restart the duplicate address detection process for the configured address when the interface changes to the Up state from the Down state.

To configure the number of NS messages sent for duplicate address detection, use the following commands.

Command	Function
Ruijie#**configure terminal**	Enters global configuration mode.
Ruijie(config)#**interface** *interface-id*	Enters interface configuration mode. Note that the **no switchport** command must be used to switch the layer-2 interface to the layer-3 interface.
Ruijie(config-if)#**ipv6 nd dad attempts** *attempts*	Configures the number of NS messages sent for duplicate address detection. When it is set to 0, the duplicate address detection function is disabled on the interface.
Ruijie(config-if)#**end**	Returns to privileged EXEC mode.
Ruijie#**show ipv6 interface vlan** *1*	Displays the IPv6 information on the interface.
Ruijie#**copy running-config startup-config**	Saves the configuration.

To restore the default value, use the **no ipv6 nd dad attempts** command.

The following example configures the number of NS messages sent for duplicate address detection on the SVI1.

```
Ruijie(config)# interface GigabitEthernet 0/1
Ruijie(config-if)# ipv6 nd dad attempts 3
Ruijie(config-if)# end
Ruijie# show ipv6 interface GigabitEthernet 0/1
Ruijie(config)# interface vlan 1
Ruijie(config-if)# ipv6 nd dad attempts 3
Ruijie(config-if)# end
Ruijie# show ipv6 interface vlan 1
Interface GigabitEthernet 0/1 is Up, ifindex: 1
address(es):
Mac Address: 00:d0:f8:00:00:01
INET6: fe80::2d0:f8ff:fe00:1 , subnet is fe80::/64
INET6: fec0:0:0:1::1 , subnet is fec0:0:0:1::/64
Joined group address(es):
ff01:1::1
ff02:1::1
ff02:1::2
ff02:1::1:ff00:1
MTU is 1500 bytes
ICMP error messages limited to one every 10 milliseconds
ICMP redirects are enabled
ND DAD is enabled, number of DAD attempts: 3
ND reachable time is 30000 milliseconds
ND advertised reachable time is 0 milliseconds
ND retransmit interval is 1000 milliseconds
ND advertised retransmit interval is 0 milliseconds
ND router advertisements are sent every 200 seconds<160--240>
ND router advertisements live for 1800 seconds
```

Configuring the IPv6 MTU

If an IPv6 packet exceeds the interface MTU size, the RGOS software splits the packet. For all devices in the same physical network segment, the IPv6 MTU of interconnected interfaces must be the same. The IPv6 MTU of interfaces automatically keeps consistent with the link MTU of interfaces.

Use the following commands to configure the IPv6 MTU.

Command	Function
Ruijie(config-if)# **ipv6 mtu** *bytes*	Configures the interface MTU value. The range is from 1,280 to 1,500.
Ruijie(config-if)# **no ipv6 mtu**	Restores the default MTU value.

Configuring Other Interface Parameters

The IPv6 parameters on an interface are divided into two parts. One is used to control the behavior of the device itself, and the other is used to control the contents of the RA sent by the device to determine what action should be taken by the host when the host receives this RA.

The following table describes these commands.

Command	**Function**
Ruijie#**configure terminal**	Enters global configuration mode.
Ruijie(config)#**interface** *interface-id*	Enters interface configuration mode. Note that the **no switchport** command must be used to switch the layer-2 interface to the layer-3 interface.
Ruijie(config-if)#**ipv6 enable**	Enables the IPv6 function.
Ruijie(config-if)#**ipv6 nd ns-interval** *milliseconds*	(Optional) Defines the retransmission interval of the NS message, in milliseconds. The default value is 1000 milliseconds.
Ruijie(config-if)#**ipv6 nd reachable-time** *milliseconds*	(Optional) Defines the time during which the neighbor is considered as reachable, in milliseconds. The default value is 30000 milliseconds.
Ruijie(config-if)# **ipv6 nd prefix** {*ipv6-prefix*/*prefix-length* | **default**} [[*valid-lifetime preferred-lifetime*] | [**at** *valid-date preferred-date*] | [**infinite** {**infinite** | *preferred-lifetime*}]] [**no-advertise**] | [[**off-link**] [**no-autoconfig**]]	(Optional) Sets the address prefix to be advertised in the RA message.
Ruijie(config-if)#**ipv6 nd ra-lifetime** *seconds*	(Optional) Sets the lifetime of the router in the RA message, namely, the time to act as the default device. The value 0 indicates that the device will not act as the default device of the directly-connected network. The default value is 1800 seconds.
Ruijie(config-if)#**ipv6 nd ra-interval** {*seconds* **min-max** *min_value max_value*}	(Optional) Sets the time interval for the device to send the RA message periodically, in seconds. The default value is 200 seconds. With **min-max** specified, the actual interval for sending the RA message is a random value between the minimum value and the maximum value. Without **min-max** specified, the actual interval for sending the RA message is the configured value multiplied by 1.2 or 0.8.
Ruijie(config-if)#**ipv6 nd managed-config-flag**	(Optional) Sets the **managed address configuration** flag bit of the RA message, and determines whether the host receiving the RA message will use the stateful auto-configuration to obtain the address. The flag bit is not configured for the RA message by default.
Ruijie(config-if)#**ipv6 nd other-config-flag**	(Optional) Sets the **other stateful configuration** flag bit of the RA message, and determines whether the host receiving the RA message will use the stateful auto-configuration to obtain information other than the address. The flag bit is not configured for the RA message by default.
Ruijie(config-if)#**ipv6 nd suppress-ra**	(Optional) Sets whether to suppress the RA message on this interface. The flag bit is not configured for the RA message by default.
Ruijie(config-if)#**end**	Returns to privileged EXEC mode.
Ruijie#**show ipv6 interface** [*interface-id*]	Displays the IPv6 information or the information of the RA sent by

Command	Function
[**ra-info**]	this interface.
Ruijie#**copy running-config startup-config**	(Optional) Saves the configuration.

To restore the default value, use the **no** commands of above commands. For details, see the *IPv6 Command Reference*.

Monitoring and Maintaining IPv6

Use the following commands to display some internal information of the IPv6 protocol, such as the IPv6 information, the neighbor table, and the routing table information of an interface.

Command	Function
show ipv6 interface [*interface-id*] [**ra-info**]	Displays the IPv6 information of the interface.
Show ipv6 neighbors [vrf *vrf-name*] [**verbose**] [*interface-id*] [*ipv6-address*]	Displays the neighbor information.
Show ipv6 route [vrf *vrf-name*] [**static** | **local | connected | bgp | rip | ospf | isis**]	Displays the information of the IPv6 routing table.

- Display the IPv6 information of an interface.

```
Ruijie# show ipv6 interface
interface GigabitEthernet 0/1 is Down, ifindex: 1
address(es):
Mac Address: 00:d0:f8:00:00:01
INET6: fe80::2d0:f8ff:fe00:1 , subnet is fe80::/64
INET6: fec0:1:1:1::1 , subnet is fec0:1:1:1::/64
Joined group address(es):
ff01:1::1
ff02:1::1
ff02:1::2
ff02:1::1:ff00:1
MTU is 1500 bytes
ICMP error messages limited to one every 10 milliseconds
ICMP redirects are enabled
ND DAD is enabled, number of DAD attempts: 1
ND reachable time is 30000 milliseconds
ND advertised reachable time is 0 milliseconds
ND retransmit interval is 1000 milliseconds
ND advertised retransmit interval is 0 milliseconds
ND router advertisements are sent every 200 seconds<160--240>
ND router advertisements live for 1800 seconds
```

- Display the information of the RA message to be sent on an interface.

```
Ruijie# show ipv6 interface ra-info
GigabitEthernet 0/1: DOWN
RA timer is stopped
waits: 0, initcount: 3
statistics: RA(out/in/inconsistent): 4/0/0, RS(input): 0
Link-layer address: 00:00:00:00:00:01
```

```
Physical MTU: 1500
ND router advertisements live for 1800 seconds
ND router advertisements are sent every 200 seconds<160--240>
Flags: !M!O, Adv MTU: 1500
ND advertised reachable time is 0 milliseconds
ND advertised retransmit time is 0 milliseconds
ND advertised CurHopLimit is 64
Prefixes: (total: 1)
fec0:1:1:1::/64(Def, Auto, vltime: 2592000, pltime: 604800, flags: LA)
```

- Display the neighbor table information of IPv6.

```
Ruijie# show ipv6 neighbors
IPv6 Address               Linklayer Addr  Interface
fe80::200:ff:fe00:1      0000.0000.0001 GigabitEthernet 0/1
State: REACH/H Age: - asked: 0
fec0:1:1:1::1          0000.0000.0001 GigabitEthernet 0/1 State: REACH/H Age: - asked: 0
```

Configuring IPv6 Tunnels

Overview

IPv6 is designed to inherit and replace IPv4. However, the evolution from IPv4 to IPv6 is a gradual process. Therefore, it is inevitable that these two protocols coexist for a period before IPv6 completely replaces IPv4. At the beginning of this transition stage, IPv4 networks are still main networks. IPv6 networks are similar to isolated islands in IPv4 networks. The problems about transition can be divided into the following two types:

1) Communication among isolated IPv6 networks via IPv4 networks
2) Communication between IPv6 networks and IPv4 networks

This article discusses the tunnel technology that is used to solve problem 1. The solution to problem 2 is Network Address Translation-Protocol Translation (NAT-PT), which is not covered in this article.

The IPv6 tunnel technology encapsulates IPv6 packets in IPv4 packets. In this way, IPv6 packets can communicate via IPv4 networks. Therefore, with the IPv6 tunnel technology, isolated IPv6 networks can communicate with each other via existing IPv4 networks, avoiding any modification and upgrade to existing IPv4 networks. An IPv6 tunnel can be configured between area border routers (ABRs) or between an ABR and a host. However, all the nodes at the two ends of the tunnel must support the IPv4 and IPv6 protocol stacks. At present, the following tunnel technologies are supported.

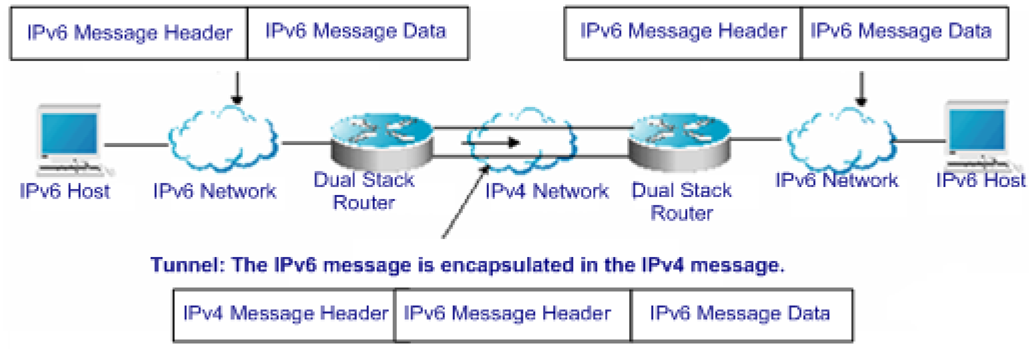
Tunnel Type	Reference
Manually Config Tunnel	RFC 2893
Automatic 6to4 Tunnel	RFC 3056
Intra-Site Automatic Tunnel Addressing Protocol(ISATAP)	draft-ietf-ngtrans-isatap-22

Caution

Interconnecting the isolated IPv6 networks through the IPv6 tunnel technology is not the ultimate IPv6 network architecture. Instead, it is a transitional technology.

The model using the tunnel technology is shown in the following figure:

Figure 11



The features of various tunnels are respectively described below.

Manually Configured IPv6 Tunnel

One manually configured tunnel is similar to one permanent link set up between two IPv6 domains via the IPv4 backbone network. It is applicable to the relatively fixed connections that have a higher requirement on security between two ABRs or between an ABR and a host.

On a tunnel interface, you must manually configure the IPv6 address, source IPv4 address (tunnel source) and destination IPv4 address (tunnel destination) of the tunnel. The nodes at the two ends of the tunnel must support the IPv6 and IPv4 protocol stacks. In practical applications, tunnels are always manually configured in pairs. You can think it as a point-to-point tunnel.

Configuring GRE Tunnel

A GRE tunnel allows a user to use a transport protocol (such as IP) to transmit network packets of any protocol. Our products support four types of GRE tunnels: IPv4 over IPv4, IPv6 over IPv4, IPv6 over IPv6, and IPv4 over IPv6.

On the tunnel interface, the IP address of the tunnel source and the IP address of the tunnel destination must be configured manually, and nodes at both ends of the tunnel must support IPv6 and IPv4 protocol stacks. The GRE tunnel is always configured simultaneously on two edge devices, and can be considered as a point-to-point tunnel.

Note

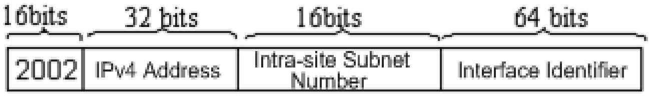
1. IPv4 over IPv6 GRE tunnel and IPv6 over IPv6 GRE tunnel are evaluation indicators.
2. IPv4 over IPv4 GRE is an evaluation indicator on the switch.
3. IPv6 over IPv4 GRE is an evaluation indicator on S5750 series switches.

Configuring Automatic 6to4 Tunnel

The automatic 6to4 tunnel technology allows isolated IPv6 networks to be interconnected via IPv4 networks. The difference between the automatic 6to4 tunnel and manually configured tunnel technologies is that the manual configured tunnel is a point-to-point tunnel, while a 6to4 tunnel is a point -to-multipoint tunnel.

The 6to4 tunnel uses an IPv4 network as a nonbroadcast multi-access (NBMA) link. Therefore, the devices of 6to4 need not be configured in pairs. The IPv4 address embedded in an IPv6 address will be used to look for the other end of the automatic tunnel. The 6to4 tunnel can be deemed as a point -to-multipoint tunnel. The automatic 6to4 tunnel can be configured on an ABR of one isolated IPv6 network. For each packet, it will automatically set up a tunnel to an ABR in another IPv6 network. The destination address of the tunnel is the IPv4 address of an ABR in the IPv6 network at the other end. The IPv4 address will be extracted from the destination IPv6 address of the packet. The destination IPv6 address begins with the prefix 2002::/16 in the following format.

Figure 12



IPv6 6to4 Address Format

The 6to4 address is an address for the automatic 6to4 tunnel technology. The IPv4 address embedded in it is usually the global IPv4 address of the egress of the ABR of the site. When the automatic tunnel is set up, the address is used as the destination IPv4 address for tunnel packet encapsulation. All the routers at the two ends of the 6ot4 tunnel must support the IPv6 and IPv4 protocol stacks. A 6to4 tunnel is usually configured between ABRs.

For example, if the global IPv4 address of the egress of the ABR of the site is 211.1.1.1 (D301:0101 in hexadecimal notation), a subnet number in the site is 1 and the interface identifier is 2e0:ddff:fee0:e0e1, then the corresponding 6to4 address can be denoted as follows:

2002: D301:0101:1: 2e0:ddff:fee0:e0e1

Caution

The IPv4 address embedded in the 6to4 address cannot be a private IPv4 address (i.e., the address of the network interface segment 10.0.0.0/8, 172.16.0.0/12 or 192.168.0.0/16) and must be the global IPv4 address.

Common application models of 6to4 tunnels:

- Simple application models

The simplest and most common application of 6to4 tunnels is used to interconnect multiple IPv6 sites. Each of the sites must have one connection to one of their shared IPv4 networks at least. This IPv4 network can be the Internet or a internal backbone network of an organization. The key is that each site must have a unique global IPv4 address. The 6to4 tunnel will use the address to form the IPv6 prefix of 6to4/48: 2002:IPV4 address/48.

- Hybrid application models

Based on the application described above, other 6to4 networks access the IPv6-only network through 6to4 relay devices at the edge. The router used to implement the function is called a 6to4 relay router.

Configuring ISATAP Automatic Tunnel

The Intra-site Automatic Tunnel Addressing Protocol (ISATAP) is a type of IPv6 tunnel technology by which an intra-site IPv6 architecture uses an IPv4 network as one nonbroadcast multi-access (NBMA) link layer, namely, using an IPv4 network as the virtual link layer of IPv6.

ISATAP is applicable to the case where the IPv6-only network inside a site is not ready for use yet and an IPv6 packet needs to be transferred internally in the site. For example, a few IPv6 hosts for test need to communicate with each other inside the site. By using an ISATAP tunnel, the IPv4/IPv6 dual stack hosts on a same virtual link can communicate with each other inside the site.

At the ISATAP site, the ISATAP device provides a standard router advertisement message, allowing the ISATAP host to be automatically configured inside the site. At the same time, the ISATAP device forwards the packets between an intra-site ISATAP host and an external IPv6 host.

The IPv6 address prefix used by ISATAP can be any legal 64-bit prefix for IPv6 unicast, including the global address prefix, link-local prefix and site-local prefix. The IPv4 address is placed as the last 32 bits of the IPv6 address, allowing a tunnel to be automatically set up.

ISATAP can be easily used with other transition technologies. Especially when used with the 6to4 tunnel technology, it can enable the dual-stack host of an intranet access an IPv6 backbone network very easily.

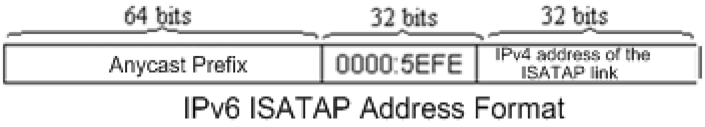
- ISATAP interface identifier

The unicast address used by ISATAP is in the form of a 64-bit IPv6 prefix plus a 64-bit interface identifier. The 64-bit interface identifier is generated in the revised EUI-64 address format. The value of the first 32 bits of the interface identifier is 0000:5EFE, an interface identifier of ISATAP.

- ISATAP address structure

An ISATAP address refers to a unicast address containing an ISATAP interface identifier in its interface identifier. An ISATAP address structure is shown in the following figure.

Figure 13



The above figure shows that the interface identifier contains an IPv4 address. The address is the IPv4 address of a dual-stack host and will be used when an automatic tunnel is automatically set up.

For example, the IPv6 prefix is 2001::/64 and the embedded IPv4 address is 192.168.1.1. In the ISATAP address, the IPv4 address is denoted as the hexadecimal numeral C0A8:0101. Therefore, its corresponding ISATAP address is as follows:

2001::0000:5EFE:C0A8:0101

Configuring 6RD Tunnel

If you want to configure a IPv6 rapid development (6RD) tunnel, configure the tunnel interface with both the source IPv4 address and destination IPv4 address. The host or device at the peer end of the tunnel should be configured in the same way.

Command	Description
Ruijie#**configure terminal**	Enters global configuration mode.

Ruijie(config)#**interface tunnel** *tunnel-number*	Configures a tunnel by specifying the tunnel number and enters interface configuration mode.
Ruijie(config-if-Tunnel id)#**tunnel mode ipv6ip 6rd**	Sets the tunnel type to 6RD tunnel,
Ruijie(config-if-Tunnel id)#**ipv6 enable**	Enables the IPv6 function on the interface. You can also enable the IPv6 function directly by configuring an IPv6 address.
Ruijie(config-if-Tunnel id)#**tunnel 6rd prefix** *ipv6-prefix* **/** *prefix-length*	Sets the 6RD prefix. If 6RD prefix is not configured, the 6RD tunnel cannot be up. If the prefix length is set to 0, it indicates that the prefix is deleted.
Ruijie(config-if-Tunnel id)#**tunnel 6rd ipv4 prefix-length** *length* **suffix-length** *length*	Sets the IPv4 prefix and suffix length for the 6RD domain. The valid range is from 0 to 31. The sum of the prefix and suffix length cannot be greater than 31. The default is 0.
Ruijie(config-if-Tunnel id)#**tunnel source** { *ipv4-address* | *interface-type interface-number* }	Sets the IPv4 source address of the tunnel or the referenced source interface number. The interface that is specified must be configured with an IPv4 address.
Ruijie(config-if-Tunnel id)#**tunnel 6rd br** *ipv4-address*	Sets the border relay (BR) IPv4 address for the 6RD customer edge (CE). This command is configured only on the 6RD CE. Configuring this command allows the 6RD router to disable security check on the tunnel packet containing this source address.
Ruijie(config-if-Tunnel id)#**exit**	Returns to privileged EXEC mode.
Ruijie(config)#**ipv6 route** *prefix::/prefix-length next-hop*	Configures the tunnel route.

The following example configures the 6RD tunnel on the device.

```
Ruijie#configure terminal
Enter configuration commands, one per line.  Exit with CNTL/Z.
Ruijie(config)#interface Tunnel 100
Ruijie(config-if-Tunnel 100)#tunnel mode ipv6ip 6rd
Ruijie(config-if-Tunnel 100)#tunnel source 10.1.1.1
Ruijie(config-if-Tunnel 100)#tunnel 6rd prefix 2001:DA8::/32
Ruijie(config-if-Tunnel 100)#tunnel 6rd ipv4 prefix-length 16 suffix-length 0
Ruijie(config-if-Tunnel 100)#tunnel 6rd br 10.1.4.1
Ruijie(config-if-Tunnel 100)#ipv6 enable
Ruijie(config)#ipv6 address 2004::1/128
Ruijie(config)#ipv6 route 2001:da8::/32 Tunnel 100
Ruijie(config)#ipv6 route ::/0 Tunnel8 2001:DA8:401::1
Ruijie(config)#ipv6 route 2001:da8:101::/48 Null 0
```

Configuring 6RD Tunnel via DHCP Automatic Configuration

You can configure the 6RD parameter for the DHCP client via the DHCP option on the DHCP server. The 6RD parameter includes the generic IPv4 prefix and suffix length, 6RD prefix length, 6RD prefix, and IPv4 address of the 6RD BR for a given 6RD domain. If you want to create a 6RD tunnel for the DHCP client, you can configure the DHCP option 212 for the client to obtain the 6RD parameter. Use the following command in DHCP address pool configuration mode to configure the 6RD parameter available for the DHCP client.

Command	Function
Ruijie(dhcp-config)# **option 6rd ipv4masklen** <*mask-length*> **ipv6prefixlen** <*prefix-length*> **ipv6prefix** <*ipv6-prefix*> **br-addr** <*ipv4-address*>	Configures the 6RD parameter.
Use the following command to enable the DHCP 6RD client interface to obtain IP address.**Command**	**Function**
Ruijie(config-if-GigabitEthernet 0/0)#ip address dhcp 6rd	Enables the DHCP client to obtain the IP address.

The following example configures the 6RD parameter for the DHCP client on the DHCP server.

```
Ruijie#configure terminal
Enter configuration commands, one per line.  Exit with CNTL/Z.
Ruijie(config)#ip dhcp pool 6rd
Ruijie(dhcp-config)#option 6rd ipv4masklen 16 ipv6prefixlen 32 ipv6prefix 2002:DA8:: br-addr
1.1.1.1
```

Configuring IPv6 Tunnels

Manually Configuring IPv6 Tunnels

This section describes how to configure tunnels manually.

To configure a tunnel manually, configure an IPv6 address on the tunnel interface and manually configure the IPv4 addresses of the source and destination of the tunnel. Then, configure the hosts or devices at the two ends of the tunnel to ensure that they support the dual stacks (the IPv6 and IPv4 protocol stacks).

Caution Do not configure tunnels manually with the same runnel source and tunnel destination.

Brief steps

```
config terminal
interface tunnel tunnel-num
tunnel mode ipv6ip
ipv6 enable
tunnel source {ip-address | type num}
tunnel destination ip-address
end
```

To configure an IPv6 tunnel manually, use the following commands in global configuration mode.

Command	Function
configure terminal	Enters global configuration mode.
interface tunnel *tunnel-num*	Specifies a tunnel interface number to create a tunnel interface and enters interface configuration mode.

Command	Function
tunnel mode **ipv6ip**	Sets the tunnel type to manually configured tunnel.
ipv6 enable	Enables the IPv6 function on the interface. You can also configure the IPv6 address to directly enable the IPv6 function on the interface.
tunnel source {*ip-address* | *type* *num*	Specifies the IPv4 source address or referenced source interface number of the tunnel. Note: If you specify an interface, the IPv4 address must have been configured on the interface.
tunnel destination *ip address*	Specifies the destination address of the tunnel.
end	Returns to privileged EXEC mode.
copy running-config startup-config	Saves the configuration information.

See the "Verifying and Monitoring IPv6 Tunnel Configuration" section to check the operation of the tunnel.

Configuring GRE Tunnels

This section describes how to configure GRE tunnels.

To configure a GRE tunnel, you need to manually configure the tunnel source IP address and tunnel destination IP address on the tunnel interface. The corresponding configurations must also be done on the peer host or device.

Caution

Do not configure a GRE tunnel with the same tunnel source IP address and tunnel destination IP address on the device.

Use the following commands to configure a GRE tunnel.

Command	Function
Ruijie# **configure terminal**	Enters global configuration mode.
Ruijie(config)# **interface tunnel** *tunnel-num*	Specifies the tunnel interface number to create a tunnel interface and enters interface configuration mode.
Ruijie(config-if-Tunnel id)# **tunnel mode** **gre** {**ip | ipv6**}	Sets the tunnel type to GRE tunnel, and specifies the carrier protocol as IPv4 or IPv6.
Ruijie(config-if-Tunnel id)# **tunnel source** {*ipv4-address* | *ipv6-address* | *interface-type* *interface-num* }	Specifies the IPv4 address of the tunnel source or source interface number referenced. If the interface is specified, the IPv4 address must have been configured on the interface.
Ruijie(config-if-Tunnel id)# **tunnel destination** *{ipv4-address* | *ipv6-address}*	Specifies the destination address of the tunnel. If the carrier protocol is IPv6, the IP address must be configured as the IPv6 address of the peer device.
Ruijie(config-if-Tunnel id)# **end**	Returns to privileged EXEC mode.
Ruijie# **copy running-config startup-config**	Saves the configuration information.

See the "Verifying and Monitoring IPv6 Tunnel Configuration" section to check the operation of the tunnel.

Caution

Because GRE features differ from device to device, the aforementioned commands may not be available on certain products.

Configuring 6to4 Tunnels

This section describes how to configure a 6to4 tunnel.

The destination address of a 6to4 tunnel is determined by the IPv4 address which is extracted from a 6to4 IPv6 address. The devices at the two ends of the 6to4 tunnel must support the dual stacks, namely, the IPv4 and IPv6 protocol stacks.

Caution

A device supports only one 6to4 tunnel. The encapsulation source address (IPv4 address) used by the 6to4 tunnel must be a globally routable address. Otherwise, the 6to4 tunnel will not work normally.

Brief steps

```
config terminal
interface tunnel tunnel-num
tunnel mode ipv6ip 6to4
ipv6 enable
tunnel source {ip-address | type num}
exit
ipv6 route 2002::/16 tunnel tunnel-number
end
```

To configure a 6to4 tunnel, use the following commands.

Command	Function
configure terminal	Enters global configuration mode.
interface tunnel *tunnel-num*	Specifies a tunnel interface number to create a tunnel interface and enters interface configuration mode.
tunnel mode ipv6ip 6to4	Sets the tunnel type to 6to4 tunnel.
ipv6 enable	Enables the IPv6 function of the interface. You can also configure the IPv6 address to directly enable the IPv6 function of the interface.
tunnel source {*ip-address* | *type num*	Specifies the encapsulation source address or referenced source interface number of the tunnel. Note: The IPv4 address must have been configured on the referenced interface. The used IPv4 address must be a globally routable address.
Exit	Returns to global configuration mode.

Command	Function
ipv6 route *2002::/16* **tunnel** *tunnel-number*	Configures a static route for the IPv6 6to4 prefix 2002::/16 and associates the output interface with the tunnel interface, i.e., the tunnel interface specified above.
End	Returns to privileged EXEC mode.
copy running-config startup-config	Saves the configuration information.

See the "Verifying and Monitoring IPv6 Tunnel Configuration" to check the operation of the tunnel.

Configuring ISATAP Tunnels

This section describes how to configure ISATAP tunnels.

On an ISATAP tunnel interface, the configuration of an ISATAP IPv6 address and the advertisement configuration of a prefix are the same as that of a common IPv6 interface. However, the address configured for an ISATAP tunnel interface must be a revised EUI-64 address. The reason is that the last 32 bits of the interface identifier in the IPv6 address are composed of the IPv4 address of the interface referenced by the tunnel source address. See the above sections for the information about ISATAP address formats.

Caution

A device supports multiple ISATAP tunnels. However, the source of each ISATAP tunnel must be different. Otherwise, there is no way to know which ISATAP tunnel a received ISATAP tunnel message belongs to.

Brief steps

```
config terminal
interface tunnel tunnel-num
tunnel mode ipv6ip isatap
ipv6 address ipv6-prefix/prefix-length eui-64
tunnel source interface-type num
no ipv6 nd suppress-ra
end
```

To configure an ISATAP tunnel, use the following commands.

Command	Function
configure terminal	Enters global configuration mode.
interface tunnel *tunnel-num*	Specifies a tunnel interface number to create a tunnel interface and enters interface configuration mode.
tunnel mode ipv6ip isatap	Sets the tunnel type to ISATAP tunnel.
ipv6 address **ipv6-prefix/prefix-length eui-64**	Configures the IPv6 ISATAP address. Be sure to use the **eui-64** keyword. In this way, the ISATAP address will be automatically generated. The address configured on an ISATAP interface must be an ISATAP address.

Command	Function
tunnel source type *num*	Specifies the source interface number referenced by the tunnel. On the referenced interface, the IPv4 address must have been configured.
no ipv6 nd suppress-ra	Sending router advertisement messages on an interface is disabled by default. You can use the command to enable the function, allowing the ISATAP host to be automatically configured.
End	Returns to privileged EXEC mode.
copy running-config startup-config	Saves the configuration information.

See the "Verifying and Monitoring IPv6 Tunnel Configuration" to check the operation of the tunnel.

Configuring the Tunnel to Support IPv6 Multicast

Currently, on the IPv6 network, both IPv6 unicast and multicast services need to be able to traverse the IPv4 network.

It is easy to configure IPv6 tunnel multicast. The tunnel interface can be configured in the same way as other common interfaces such as an SVI interface.

Caution

Multicast is supported only in the manually configured IPv6 tunnel. For tunnels of other types, multicast can be configured, but multicast data cannot be received or forwarded after the configuration.

For the manually configured IPv6 tunnel, if the tunnel is created based on an IPv6 tunnel of any type, multicast can be configured, but multicast data cannot be received or forwarded after the configuration.

When IPv6 multicast data traverses the IPv4 network, the MTU restrictions are the same as those for IPv6 unicast data.

Verifying and Monitoring IPv6 Tunnel Configuration

This section describes how to verify the configuration and operation of an IPv6 tunnel.

Brief steps

```
enable
show interface tunnel number
show ipv6 interface tunnel mumber
ping protocol destination
show ip route
show ipv6 route
```

To verify the configuration and operation of a tunnel, use the following commands.

Command	Function
show interface tunnel *tunnel-num*	Displays the information of a specified tunnel interface.

Command	Function
show ipv6 interface tunnel *tunnel-num*	Displays the IPv6 information of the tunnel interface.
ping protocol destination	Checks the basic connectivity of a network.
show ip route	Displays the IPv4 routing table.
show ipv6 route	Displays the IPv6 router table.

3) Display the information of a tunnel interface.

```
Ruijie# show interface tunnel 1
Tunnel 1 is up, line protocol is Up
Hardware is Tunnel, Encapsulation TUNNEL
Tunnel source 192.168.5.215  ,  destination 192.168.5.204
Tunnel protocol/transport IPv6/IP
Tunnel TTL is 9
Tunnel source do conformance check set
Tunnel source do ingress filter set
Tunnel destination do safety check not set
Tunnel disable receive packet not set
```

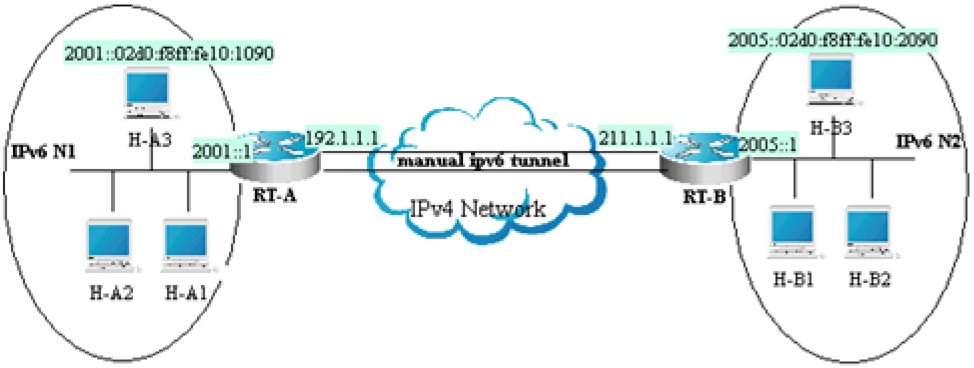
4) Display the IPv6 information of a tunnel interface.

```
Ruijie# show ipv6 interface tunnel 1
interface Tunnel 1 is Up, ifindex: 6354
address(es):
Mac Address: N/A
INET6: fe80::3d9a:1601 , subnet is fe80::/64
Joined group address(es):
ff02::2
ff01::1
ff02::1
ff02::1:ff9a:1601
INET6: 3ffe:4:0:1::1 , subnet is 3ffe:4:0:1::/64
Joined group address(es):
ff02::2
ff01::1
ff02::1
ff02::1:ff00:1
MTU is 1480 bytes
ICMP error messages limited to one every 100 milliseconds
ICMP redirects are enabled
ND DAD is enabled, number of DAD attempts: 1
ND reachable time is 30000 milliseconds
ND advertised reachable time is 0 milliseconds
ND retransmit interval is 1000 milliseconds
ND advertised retransmit interval is 0 milliseconds
ND router advertisements are sent every 200 seconds<240--160>
ND router advertisements live for 1800 seconds
```

IPv6 Tunnel Configuration Examples

Example of Configuring IPv6 Tunnels Manually

Figure 14



As shown in the above figure, IPv6 networks N1 and N2 are isolated by the IPv4 network. Now, the two networks are interconnected by configuring a tunnel manually. For example, the H-A3 host in N1 can access the H-B3 host in N2.

In the figure, RT-A and RT-B are routers that support the IPv4 and IPv6 protocol stacks. Tunnel configuration is performed on the ABRs (RT-A and RT-B) in N1 and N2. Note that the tunnel must be configured manually in pairs, that is, on RT-A and RT-B.

The following presents the tunnel configuration on routers:

Prerequisite: Assume that the routes of IPv4 are connected. In the following content, no more route configuration condition about IPv4 is listed.

RT-A:

#Connect the interfaces of the IPv4 network.

```
interface FastEthernet 2/1
no switchport
ip address 192.1.1.1 255.255.255.0
```

#Connect the interfaces of the IPv6 network.

```
interface FastEthernet 2/2
no switchport
ipv6 address 2001::1/64
no ipv6 nd suppress-ra  (optional)
```

#Configure the manual tunnel interface.

```
interface Tunnel 1
tunnel mode ipv6ip
ipv6 enable
tunnel source FastEthernet 2/1
tunnel destination 211.1.1.1
```

#Configure the route to the tunnel.

```
ipv6 route 2005::/64 tunnel 1
```

RT-B:

#Connect the interfaces of the IPv4 network.

```
interface FastEthernet 2/1
no switchport
ip address 211.1.1.1 255.255.255.0
```

# Connect the interfaces of the IPv6 network.

```
interface FastEthernet 2/2
no switchport
ipv6 address 2005::1/64
no ipv6 nd suppress-ra  (optional)
```

#Configure the manual tunnel interface.

```
interface Tunnel 1
tunnel mode ipv6ip
ipv6 enable
tunnel source FastEthernet 2/1
tunnel destination 192.1.1.1
```

#Configure the route to the tunnel.

```
ipv6 route 2001::/64 tunnel 1
```

Example of Manually Configuring IPv6 Tunnels to Support Multicast

Assume that the network topology is shown in Figure 4. On the basis of the previous example, the additional support to PIM SMv6 multicast is required. Detailed configurations related to multicast are shown below:

- RT-A

# Globally enable multicast.

```
ipv6 multicast-routing
```

# Enable PIM SMv6 on the interface.

```
interface Tunnel 1
IPv6 pim sparse-mode
```

- RT-B

# Globally enable multicast.

```
ipv6 multicast-routing
```

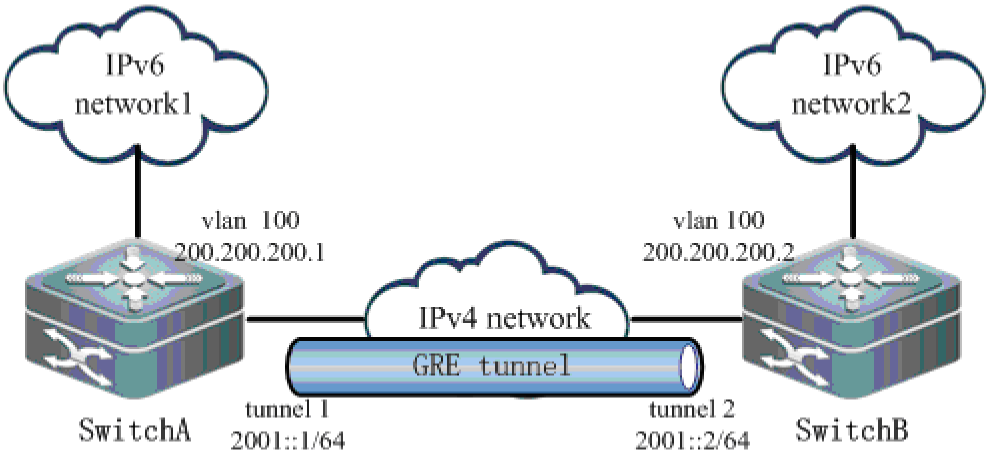
# Enable PIM SMv6 on the interface.

```
interface Tunnel 1
```

```
IPv6 pim sparse-mode
```

Example of Configuring IPv6 over IPv4 GRE Tunnels

Figure 15



As shown in Figure 15, two IPv6 networks, IPv6 network1 and IPv6 network2, need to be connected via a public IPv4 network to realize intercommunication. Layer-3 devices Switch A and Switch B supports both IPv4 and IPv6 stacks, and are interconnected via the IPv4 network. An IPv6 over IPv4 GRE tunnel needs to be created over this IPv4 network.

Assuming that IPv6 network1 is 2002::/64 and IPv6 network2 is 2003::/64, the configurations on Switch A and Switch B are shown below:

5) Configure Switch A.

# Configure interface vlan 100.

```
interface vlan 100
ip address 200.200.200.1 255.255.255.0
```

# Configure interface Tunnel 1.

```
interface Tunnel 1
 ipv6 address 2001::1/64
 tunnel mode gre ip
 tunnel source vlan 100
 tunnel destination 200.200.200.2
```

# Configure the route to pass through the tunnel interface to reach IPv6 network2.

```
ipv6 route 2003::/64 tunnel 1 2001::2
```

6) Configure Switch B.

# Configure interface vlan 100.

```
interface vlan 100
 ip address 200.200.200.2 255.255.255.0
```

# Configure interface Tunnel 1.

```
interface Tunnel1
 ipv6 address 2001::2/64
 tunnel mode gre ip
 tunnel source vlan 100
 tunnel destination 200.200.200.1
```

# Configure the route to pass through the tunnel interface to reach IPv6 network1.

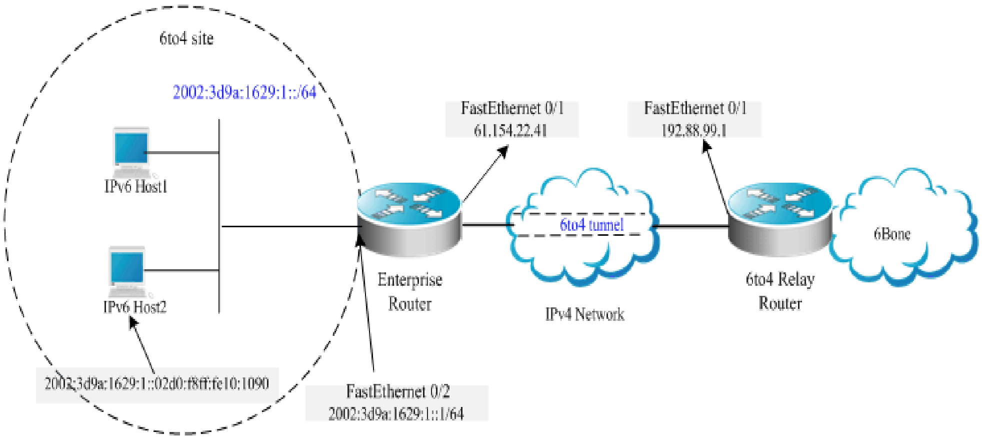
```
ipv6 route 2002::/64 tunnel 1 2001::1
```

7) View the operation of the tunnel, taking Switch A as an example.

```
show interface tunnel 1
Index(dec):3 (hex):3
Tunnel 1 is UP  , line protocol is UP
Hardware is  Tunnel
Interface address is: no ip address
  MTU 1496 bytes, BW 9 Kbit
  Encapsulation protocol is Tunnel, loopback not set
  Keepalive set (10 sec), retries 3
  Carrier delay is 2 sec
  RXload is 1 ,Txload is 1
  Tunnel source 200.200.200.1 (VLAN 100), destination 200.200.200.2
  Tunnel TOS 0x14,  Tunnel TTL 255
  Tunnel protocol/transport GRE/IP
   Key disabled, Sequencing disabled
   Checksumming of packets disabled
  Path MTU Discovery, ager 10 mins, min MTU 92, MTU 0, expires never
  Queueing strategy: FIFO
    Output queue 0/40, 0 drops;
    Input queue 0/75, 0 drops
  5 minutes input rate 0 bits/sec, 0 packets/sec
  5 minutes output rate 0 bits/sec, 0 packets/sec
    0 packets input, 0 bytes, 0 no buffer, 0 dropped
    Received 0 broadcasts, 0 runts, 0 giants
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 abort
    0 packets output, 0 bytes, 0 underruns , 0 dropped
    0 output errors, 0 collisions, 0 interface resets
```

Example of Configuring 6to4 Tunnels

Figure 16



As shown in the above figure, an IPv6 network (6to4 site) uses a 6to4 tunnel to access the IPv6 backbone network (6bone) via a 6to4 relay router.

As described above, the 6to4 tunnel technology is used to interconnect isolated IPv6 networks and the IPv6 backbone network can be accessed via the 6to4 relay router very easily. The 6to4 tunnel is an automatic tunnel and the IPv4 address embedded in the IPv6 address will be used to look for the other end of the automatic tunnel. Therefore, you need not configure the destination end for the 6to4 tunnel. Additionally, unlike a manual tunnel, the 6to4 tunnel need not be configured in pairs.

61.154.22.41 is 3d9a:1629 in hexadecimal notation.

192.88.99.1 is c058:6301 in hexadecimal notation.

⚠ Caution

When configuring a 6to4 tunnel on an ABR, be sure to use a globally routable IPv4 address. Otherwise, the 6to4 tunnel will not work normally.

The following is the configuration of the two routers in the figure (Assume that IPv4 routes are connected. Ignore the configuration of IPv4 routes.):

Enterprise router:

# Connect the interfaces of the IPv4 network.

```
interface FastEthernet 0/1
no switchport
ip address 61.154.22.41 255.255.255.128
```

# Connect the interfaces of the IPv6 network.

```
interface FastEthernet 0/2
no switchport
ipv6 address 2002:3d9a:1629:1::1/64
no ipv6 nd suppress-ra
```

# Configure the 6to4 tunnel interface.

```
interface Tunnel 1
tunnel mode ipv6ip 6to4
```

```
ipv6 enable
tunnel source FastEthernet 0/1
```

# Configure the route to the tunnel.

```
ipv6 route 2002::/16 Tunnel 1
```

# Configure the route to the 6to4 relay router to access 6bone.

```
ipv6 route ::/0 2002:c058:6301::1
```

ISP 6to4 relay router:

# Connect the interface of the IPv4 network.

```
interface FastEthernet 0/1
no switchport
ip address 192.88.99.1 255.255.255.0
```

# Configure the 6to4 tunnel interface.

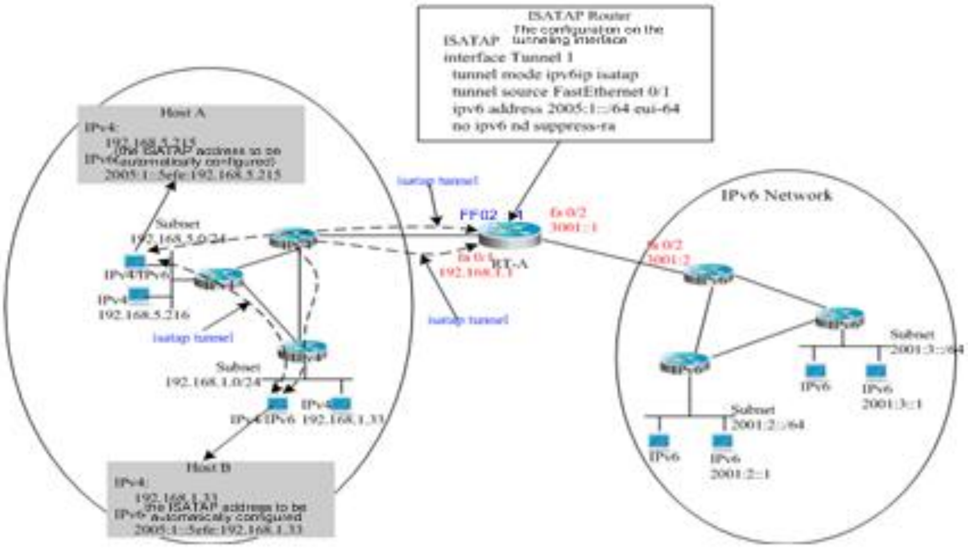
```
interface Tunnel 1
tunnel mode ipv6ip 6to4
ipv6 enable
tunnel source FastEthernet 0/1
```

# Configure the route to the tunnel.

```
ipv6 route 2002::/16 Tunnel 1
```

Example of Configuring ISATAP Tunnels

Figure 17



The above figure is one typical topology using an ISATAP tunnel. The ISATAP tunnel is used to communicate between isolated IPv4/IPv6 dual-stack hosts inside the IPv4 site. The ISATAP router has the two following functions inside the ISATAP site:

- Receive a router solicitation message from the ISATAP host inside the site and then respond with a router advertisement message for the ISATAP host inside the site to be automatically configured.
- Be responsible for the message forwarding function of the ISATAP host inside the site and the IPv6 host outside the site.

In the above figure, when Host A and Host B send the router solicitation message to the ISATAP router, the ISATAP router will respond with a router advertisement message. After receiving the message, the hosts will automatically perform configuration and generate their own ISATAP addresses respectively. Then, the IPv6 communication between Host A and Host B will be done via the ISATAP tunnel. When Host A or Host B need to communicate with the IPv6 host outside the site, Host A sends the message to the ISATAP router RT-A via the ISATAP tunnel and then RT-A forwards the message to the IPv6 network.

In the above figure, the ISATAP router (RT-A) is configured as follows:

# Connect the interfaces of the IPv4 network.

```
interface FastEthernet 0/1
no switchport
ip address 192.168.1.1 255.255.255.0
```

# Configure the ISATAP tunnel interface.

```
interface Tunnel 1
tunnel mode ipv6ip isatap
tunnel source FastEthernet 0/1
ipv6 address 2005:1::/64 eui-64
no ipv6 nd suppress-ra
```

# Connect the interfaces of the IPv6 network.

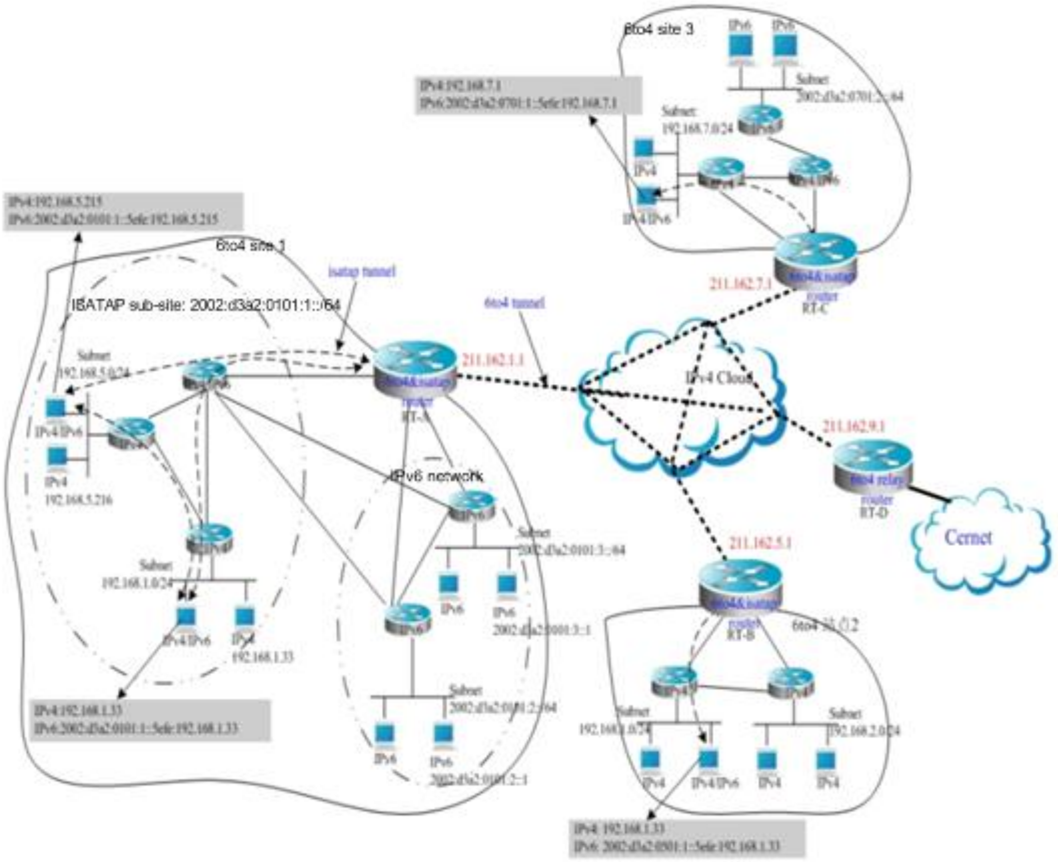
```
interface FastEthernet  0/2
no switchport
ipv6 address 3001::1/64
```

# Configure the route to the IPv6 network.

```
ipv6 route 2001::/64 3001::2
```

Example of Configuring ISATAP and 6to4 Tunnels

Figure 18



Note

The above figure shows a hybrid application of a 6to4 tunnel and an ISATAP tunnel. By using the 6to4 tunnel technology, various 6to4 sites are interconnected and the 6to4 sites access the Cernet network via the **6to4 relay router**. At the same time, by using the ISATAP tunnel technology inside the 6to4 sites, the IPv6 hosts isolated by IPv4 inside the sites perform IPv6 communication via the ISATAP tunnel.

Caution

In the above figure, the used global IP addresses including the address of the 6to4 relay router are only for convenience. When actually planning topologies, you should use a true global IP address and the address of the 6to4 relay. At present, many organizations provide the addresses of open and free 6to4 relay routers.

The configurations of ABRs at the 6to4 sites shown in the above figure are described respectively below. Note that only main related configurations are listed here.

RT-A:

# Connect the interfaces of the Internet.

```
interface GigabitEthernet 0/1
no switchport
ip address 211.162.1.1 255.255.255.0
```

# Connect the interfaces of the IPv4 network inside the site.

```
interface FastEthernet  0/1.
```

```
no switchport
ip address 192.168.0.1 255.255.255.0
```

# Configure the ISATAP tunnel interface.

```
interface Tunnel 1
tunnel mode ipv6ip isatap
tunnel source FastEthernet 0/1
ipv6 address 2002:d3a2:0101:1::/64 eui-64
no ipv6 nd suppress-ra
```

# Connect interface 1 of the IPv6 network.

```
interface FastEthernet 0/2
no switchport
2002:d3a2:0101:10::1/64
```

# Connect interface 2 of the IPv6 network.

```
interface FastEthernet  0/2
no switchport
2002:d3a2:0101:20::1/64
```

# Configure the 6to4 tunnel interface.

```
interface Tunnel 2
tunnel mode ipv6ip 6to4
ipv6 enable
tunnel source GigabitEthernet 0/1
```

# Configure the route to the 6to4 tunnel.

```
ipv6 route 2002::/16 Tunnel 2
```

# Configure the route to the 6to4 relay router RT-D to access the Cernet network.

```
ipv6 route ::/0 2002:d3a2::0901::1
```

RT-B:

# Connect the interfaces of the Internet.

```
interface GigabitEthernet 0/1
no switchport
ip address 211.162.5.1 255.255.255.0
```

# Connect interface 1 of the IPv4 network inside the site.

```
interface FastEthernet  0/1
no switchport
ip address 192.168.10.1 255.255.255.0
```

# Connect interface 2 of the IPv4 network inside the site.

```
interface FastEthernet  0/2
```

```
no switchport
ip address 192.168.20.1 255.255.255.0
```

# Configure the ISATAP tunnel interface.

```
tunnel mode ipv6ip isatap
tunnel source FastEthernet 0/1
ipv6 address 2002:d3a2:0501:1::/64 eui-64
no ipv6 nd suppress-ra
```

# Configure the 6to4 tunnel interface.

```
interface Tunnel 2
tunnel mode ipv6ip 6to4
ipv6 enable
tunnel source GigabitEthernet 0/1
```

# Configure the route to the 6to4 tunnel.

```
ipv6 route 2002::/16 Tunnel 2
```

# Configure the route to the 6to4 relay router RT-D to access the Cernet network.

```
ipv6 route ::/0 2002:d3a2::0901::1
```

RT-C:

# Connect the interfaces of the Internet.

```
interface GigabitEthernet 0/1
no switchport
ip address 211.162.7.1 255.255.255.0
```

# Connect the interfaces of the IPv4 network inside the site.

```
interface FastEthernet 0/1
no switchport
ip address 192.168.0.1 255.255.255.0
```

# Configure the ISATAP tunnel interface.

```
interface Tunnel 1
tunnel mode ipv6ip isatap
tunnel source FastEthernet 0/1
ipv6 address 2002:d3a2:0701:1::/64 eui-64
no ipv6 nd suppress-ra
```

# Connect the interfaces of the IPv6 network.

```
interface FastEthernet 0/2
no switchport
2002:d3a2:0701:10::1/64
```

# Configure the 6to4 tunnel interface.

```
interface Tunnel 2
tunnel mode ipv6ip 6to4
ipv6 enable
tunnel source GigabitEthernet 0/1
```

# Configure the route to the 6to4 tunnel.

```
ipv6 route 2002::/16 Tunnel 2
```

#Configure the route to the 6to4 relay router RT-D to access the Cernet network.

```
ipv6 route ::/0 2002:d3a2::0901::1
```

RT-D (6to4 relay):

# Connect the interfaces of the Internet.

```
interface GigabitEthernet 0/1
no switchport
ip address 211.162.9.1 255.255.255.0
```

# Connect the interfaces of the IPv6 network.

```
interface FastEthernet 0/1
no switchport
2001::1/64
no ipv6 nd suppress-ra
```

# Configure the 6to4 tunnel interface.

```
interface Tunnel 1
tunnel mode ipv6ip 6to4
ipv6 address 2002:d3a2::0901::1/64
tunnel source GigabitEthernet 0/1
```

#Configure the route to the 6to4 tunnel.

```
ipv6 route 2002::/16 Tunnel 1
```

Configuring NAT-PT

Overview

With the rapid development of the Internet, IPv4 becomes inadequate for the Internet, and IPv6 deployment has been on the agenda. To implement an IPv6 network, you are advised make the most of the existing network environment to build the next generation of the Internet to achieve smooth evolution and avoid excessive investment waste. As the Internet is currently based on IPv4 and unlikely to completely evolve to the IPv6 network within a short time, IPv6 and IPv4 networks will co-exist for quite some time.

That brings a challenge to maintain services and functions of the existing network and achieve transparent transmission between IPv4 and IPv6 networks at a low cost. Network Address Translation-Protocol Translation (NAT-PT) emerges in response to the communication between directly connected IPv6 and IPv4 networks.

Free communication between IPv6 and IPv4 networks can be ensured by using NAT-PT. Communication can be initiated by a host on either network. You can use NAT-PT to translate protocols and semantics without transforming or upgrading the host.

Basic Concepts

NAT for the IPv4 network is adopted and improved for IPv4-IPv6 address translation. It aims to establish and maintain address mappings.

PT is responsible for IPv4-IPv6 protocol translation. It builds new packets by replacing the IPv6 header with the IPv4 header or vice versa. Only certain types of the Internet Control Message Protocol version 4 (ICMPv4) and ICMPv6 packets can be translated because of their protocols, for example, translation between ICMPv4 and ICMPv6 request/response packets and translation between ICMPv4 and ICMPv6 destination-unreachable packets.

As a mode of dynamic address translation, Port Address Translation (PAT) is used for IPv6-IPv4 dynamic translation. Multiple IPv6 addresses can be mapped to the same IPv4 address differentiated by different ports to avoid IPv4 address depletion.

Working Principle

NAT-PT works on the border router between IPv6 and IPv4 networks. The NAT-PT module translates IP header addresses between IPv6 and IPv4 networks, and translates semantics of groups according to different protocols to achieve transparent transmission between IPv4 and IPv6 networks. NAT-PT can replace addresses statically or use an address pool containing global addresses. When a session passes through the border router, the border router takes an address from the address pool and assigns the address to the correct IPv4/IPv6 host. At the same time, to track the session to be translated, the session must pass though the same NAT-PT border router.

Protocols and Standards

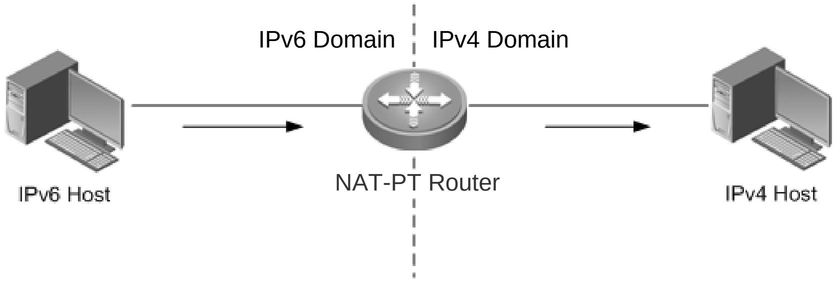
RFC2765: Stateless IP/ICMP Translation Algorithm (SIIT).

RFC2766: Network Address Translation - Protocol Translation (NAT-PT).

Applications

Scenario 1

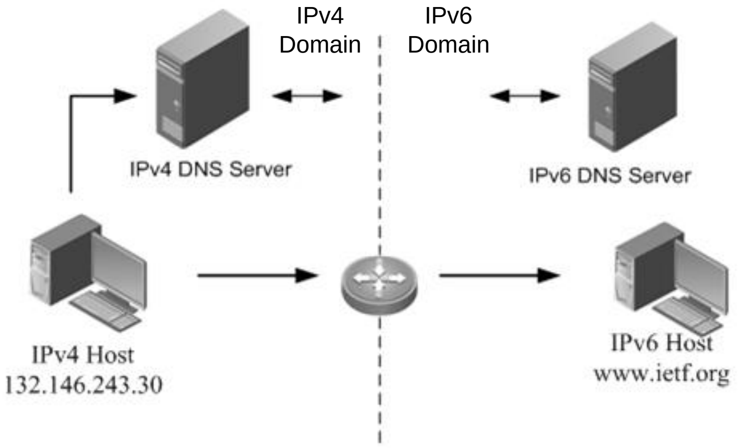
Figure 0–1 IP Address-based Access



In this scenario, NAT-PT aims at interworking between the IPv6 host and the IPv4 host to implement transparent transmission between IPv6 and IPv4 networks at a low cost without changing the topology or incurring extra expenses during packet transmission.

Scenario 2

Domain Name-based Access



In this scenario, as the IPv4 DNS packets and IPv6 DNS packets have different formats, the IPV6 DNS server cannot identify DNS requests from the IPv4 host and thereby cannot perform DNS resolution. NAT-PT is used to enable domain name-based access to the destination host, so that the IPv6 DNS server can respond to DNS requests from the IPv4 host and that domain name-based transparent transmission can be implemented between the IPv4 host and the IPv6 host.

Configuring NAT-PT

Defaults

Feature	Default Settings
NAT-PT	Disabled.
Maximum NAT-PT entry	100K

Feature	Default Settings
NAT-PT timeout	300 seconds
dns-timeout	300 seconds
finrst-timeout	60 seconds
icmp-timeout	90 seconds
syn-timeout	60 seconds
tcp-timeout	24 hours
udp-timeout	300 seconds

Configuring Static Source Address-based NAT-PT Mapping

Command	Description
Ruijie>**enable**	Enters privileged EXEC mode.
Ruijie#**configure terminal**	Enters global configuration mode.
Ruijie(config)#**ipv6 nat prefix***ipv6-prefix/prefix-length*	Assigns an IPv6 prefix as the global NAT-PT prefix.
Ruijie(config)#**interface** *interface-name interface-number*	Specifies an IPv6 interface and enters interface configuration mode.
Ruijie(config-if)#**ipv6 address** *ipv6-address/prefix-length*	Configures the IPv6 address of the interface.
Ruijie(config-if)#**ipv6 nat**	Enables NAT-PT for the interface.
Ruijie(config-if)#**exit**	Returns to global configuration mode.
Ruijie(config)#**interface** *interface-name interface-number*	Specifies an IPv4 interface and enters interface configuration mode.
Ruijie(config-if)#**ip address** *ip-address mask*	Configures the IPv4 address of the interface.
Ruijie(config-if)#**ipv6 nat**	Enables NAT-PT for the interface.
Ruijie(config-if)#**exit**	Returns to global configuration mode.
Ruijie(config)#**ipv6nat v6v4 source** *ipv6-address ipv4-address*	Configures the static IPv6-IPv4 address mapping.
Ruijie(config)#**ipv6 nat v4v6 source** *ipv4-address ipv6-address*	Configures the static IPv4-IPv6 address mapping.

Example: Configuring static source address-based NAT-PT mappings

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)#interface fastethernet 0/1
Ruijie(config-if)#ipv6 nat
RouterB(config-if)#ipv6 address 2001::1/64
RouterB(config-if)#exit
RouterB(config)#ipv6 nat v4v6 source 8.0.0.2 2001:DA8:1::5
RouterB(config)#ipv6 nat v6v4 source 2001::18.0.0.5
```

Configuring Dynamic Source Address-based NAT-PT Mappings

Command	Description
Ruijie>**enable**	Enters privileged EXEC mode.
Ruijie#**configure terminal**	Enters global configuration mode.
Ruijie(config)#**ipv6 nat prefix***ipv6-prefix/prefix-length*	Assigns an IPv6 prefix as the global NAT-PT prefix.
Ruijie(config)#**interface** *interface-name interface-number*	Specifies an IPv6 interface and enters interface configuration mode.
Ruijie(config-if)#**ipv6 address** *ipv6-address/prefix-length*	Configures the IPv6 address of the interface.
Ruijie(config-if)#**ipv6 nat**	Enables NAT-PT for the interface.
Ruijie(config-if)#**exit**	Returns to global configuration mode.
Ruijie(config)#**interface** *interface-name interface-number*	Specifies an IPv4 interface and enters interface configuration mode.
Ruijie(config-if)#**ip address** *ip-address mask*	Configures the IPv4 address of the interface.
Ruijie(config-if)#**ipv6 nat**	Enables NAT-PT for the interface.
Ruijie(config-if)#**exit**	Returns to global configuration mode.
Ruijie(config)#**ipv6 nat v6v4 source list***access-list-name***pool***v4pool-name* or Ruijie(config)#**ipv6 nat v4v6 source list** *access-list-name***pool***v6pool-name*	Enables the dynamic IPv6-IPv4 or IPv4-IPv6 address mapping. Takes an address from *v4pool-name* (name of the IPv6 address pool) for translation to match the IPv6 Access Control List (ACL) entry **access-list-name**; or takes an address from *v6pool-name* (name of the IPv4 address pool) for translation to match the IPv4 ACL entry **access-list-name**.
Ruijie(config)#**ipv6 nat v6v4 pool** *v4pool-name start-ipv4 end-ipv4***prefix-length** *prefix-length* or Ruijie(config)#**ipv6 nat v4v6 pool***v6pool-namestart-ipv6 end-ipv6***prefix-length** *prefix-length*	Configures the IPv4 address pool with a specified prefix length or configures the IPv6 address pool with a specified prefix length.
Ruijie(config)#**ipv6 access-list** *access-list-name* or Ruijie(config)#**ip access-list**{ **standard** | **extended** } { *id* | *access-list-name* }	Configures IPv6 or IPv4 ACL Permit entries.

Example: Configuring dynamic source address-based NAT-PT mappings

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)#interface fastethernet 0/1
Ruijie(config-if)#ipv6 nat
RouterB(config-if)#exit
RouterB(config)#ipv6 nat v6v4 source list v6_acl pool v4pool
RouterB(config)#ipv6 nat v6v4 pool v4pool 20.0.0.1 20.0.0.30 prefix-length 24
RouterB(config)#ipv6 nat v4v6 source 8.0.0.1 2001:DA8:1::5
RouterB(config)#ipv6 access-list v6_acl
RouterB(config-ipv6-acl)#permit ipv6 2001:DA8:2::/64 any
```

Monitoring

Command	Description
Ruijie#**clear ipv6 nat statistics**	Clears all NAT-PT statistics.
Ruijie#**clear ipv6 nat translations ***	Clears all records of NAT-PT.
Ruijie#**clear ipv6 nat translations icmp**	Clears all records of NAT-PT ICMP.
Ruijie#**clear ipv6 nat translations tcp**	Clears all records of NAT-PT Transmission Control Protocol (TCP).
Ruijie#**clear ipv6 nat translations udp**	Clears all records of NAT-PT User Datagram Protocol (UDP).
Ruijie#**debug ipv6 nat**	Displays NAT-PT debugging information.
Ruijie#**show ipv6 nat memory**	Displays NAT-PT memory information.
Ruijie#**show ipv6 nat pool**	Displays NAT-PT address pools.
Ruijie#**show ipv6 nat rule**	Displays translation rules.
Ruijie#**show ipv6 nat statistics**	Displays NAT-PT statistics.
Ruijie#**show ipv6 nat translations**	Displays records of NAT-PT.

Configuration Example

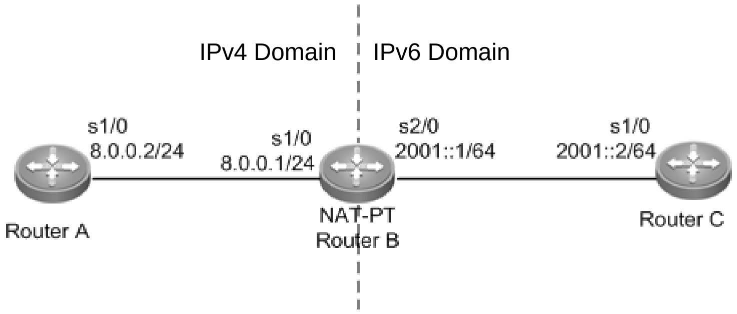
Static NAT-PT

Networking Requirements

Router C whose IPv6 address is 2001::2/64 wants to interwork with Router A whose IPv4 address is 8.0.0.2/24. For this purpose, you need to deploy Router B as an NAT-PT device between the IPv4 domain and the IPv6 domain and configure static IPv4 and IPv6 packet mappings on Router B to implement interworking between the IPv4 domain and the IPv6 domain.

Network Topology

Static NAT-PT Configuration



Configuration Tips

The following describes the Configuration Tips of configuration for static NAT-PT mappings:

- Configure an IPv4 address.
- Configure an IPv6 address.
- Configure the NAT-PT prefix and enable NAT-PT.
- Configure static IPv4 and IPv6 packet mappings.
- Configure the static route for IPv4 or IPv6.

Caution

When configuring the static source address-based mapping on Router C, you must avoid confliction between the mapped source IPv4 address and other addresses in the IPv4 domain to ensure a reachable route from Router A to the NAT-PT device (Router B).

Steps

- Configure the IP address of Router A in the IPv4 domain.

```
RouterA>enable
RouterA#configure terminal
RouterA(config)#interface s1/0
RouterA(config-if)#ip address 8.0.0.2 255.255.255.0
```

- Configure the IP address of Router C in the IPv6 domain.

```
RouterC>enable
RouterC#configure terminal
RouterC(config)#interface s1/0
RotuerC(config-if)#ipv6 enable
RouterC(config-if)#ipv6 address 2001::2/64
```

- Configure Router B.

```
RouterB>enable
RouterB#configure terminal
RouterB(config)#ipv6 nat prefix 2001:DA8:1::/96
RouterB(config)#interface s1/0
RouterB(config-if)#ip address 8.0.0.1 255.255.255.0
RouterB(config-if)#ipv6 nat
RouterB(config-if)#exit
RouterB(config)#interface s2/0
RouterB(config-if)#ipv6 enable
RouterB(config-if)#ipv6 nat
RouterB(config-if)#ipv6 address 2001::1/64
RouterB(config-if)#exit
RouterB(config)#ipv6 nat v4v6 source 8.0.0.2 2001:DA8:1::5
RouterB(config)#ipv6 nat v6v4 source 2001::2 8.0.0.5
```

- On Router C, configure a static route destined for the network segment matched to the NAT-PT prefix.

```
RouterC>enable
RouterC#configure terminal
RouterC(config)#ipv6 route 2001:DA8:1::/96 2001::1
```

Verification

```
Ruijie#show ipv6 nat translations

Prot   IPv4 source                    IPv6 source

       IPv4 destination               IPv6 destination

---    8.0.0.52001::2

       ---                            ---

---    ---                            ---

8.0.0.2                       2001:DA8:1::1

icmp   8.0.0.2,47             2001:DA8:1::5,47

8.0.0.5,47                    2001::2,47
```

Run the **ping 8.0.0.5** command on Router A to receive response packets.

```
Ruijie#ping 8.0.0.5

Sending 5, 100-byte ICMP Echoes to 8.0.0.5, timeout is 2 seconds:

< press Ctrl+C to break >

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/10 ms
```

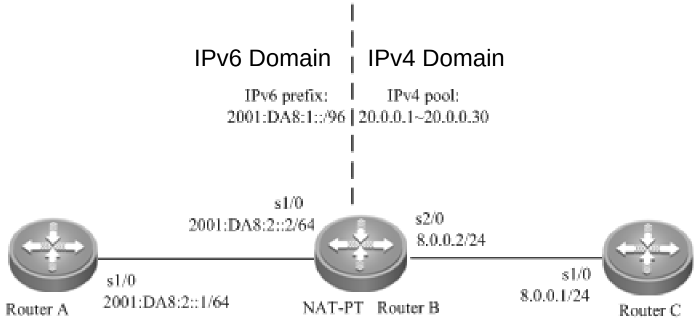
Dynamic NAT-PT

Networking Requirements

Router A whose IPv6 address is 2001:DA8:2::1 wants to access Router C whose IPv4 address is 8.0.0.1, but Router C is forbidden to access Router A. For this purpose, you need to deploy Router B between the IPv6 domain and the IPv4 domain for dynamic NAT-PT.

Network Topology

Dynamic NAT-PT configuration



Configuration Tips

The following describes the Configuration Tips of configuration for dynamic NAT-PT:

- Configure an IPv4 address.
- Configure an IPv6 address.
- Configure the NAT-PT prefix and enable NAT-PT.
- Configure dynamic address mapping in the IPv6 domain and static address mapping in the IPv4 domain.
- Configure the IPv4 address pool with a specified length.
- Configure IPv6 ACL Permit entries.

Steps

- Configure the IP address of Router C in the IPv4 domain.

```
RouterA>enable
RouterA#configure terminal
RouterA(config)#interface s1/0
RouterA(config-if)#ip address 8.0.0.1 255.255.255.0
```

- Configure Router B.

```
RouterB>enable
RouterB#configure terminal
RouterB(config)#ipv6 nat prefix 2001:DA8:1::/96
RouterB(config)#interface s1/0
RouterB(config-if)#ipv6 enable
RouterB(config-if)#ipv6 address 2001:DA8:2::2/64
RouterB(config-if)#ipv6 nat
RouterB(config-if)#exit
RouterB(config)#interface s2/0
RouterB(config-if)#ipv6 address 8.0.0.2 255.255.255.0
RouterB(config-if)#ipv6 nat
RouterB(config-if)#exit
RouterB(config)#ipv6 nat v6v4 source list v6_acl pool v4pool
RouterB(config)#ipv6 nat v6v4 pool v4pool 20.0.0.1 20.0.0.30 prefix-length 24
RouterB(config)#ipv6 nat v4v6 source 8.0.0.1 2001:DA8:1::5
RouterB(config)#ipv6 access-list v6_acl
RouterB(config-ipv6-acl)#permit ipv6 2001:DA8:2::/64 any
```

- Configure the IP address of Router A in the IPv6 domain.

```
RouterC>enable
RouterC#configer terminal
RouterC(config)#interface s1/0
RotuerC(config-if)#ipv6 enable
RouterC(config-if)#ipv6 address 2001:DA8:2::1/64
```

- On Router A, configure the static route destined for the network segment matched to the NAT-PT prefix.

```
RouterC>enable
RouterC#configure terminal
RouterC(config)#ipv6 route 2001:DA8:1::/96 2001:DA8:2::2
```

- On Router C, configure the static route destined for the network segment 20.0.0.0/24.

```
RouterC>enable
RouterC#configure terminal
RouterC(config)#ip route 20.0.0.0 255.255.255.08.0.0.2
```

Verification

```
Ruijie#show ipv6 nat translations
Prot  IPv4 source            IPv6 source
     IPv4 destination        IPv6 destination
---   8.0.0.12001:DA8:1::5
     ---                  ---
icmp  20.0.0.6,1024       2001:DA8:2::1,1024
8.0.0.1,10242001:DA8:1::5,1024
udp20.0.0.6,532001:DA8:2::1,53
8.0.0.1,13642001:DA8:1::5,1364
```

Run the **ping 2001:DA8:1::5** command on Router A to receive response packets from Router C.

```
Ruijie#ping 2001:DA8:1::5
Sending 5, 100-byte ICMP Echoes to 2001:DA8:1::5, timeout is 2 seconds:
< press Ctrl+C to break >
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/10 ms
```

Configuring Stateful NAT64

Understanding Stateful NAT64

Overview

With the fast development of the Internet, IPv4 can no longer meet Internet requirements. Under this circumstance, IPv6 is about to be deployed. To support an IPv6 network, you must make full use of existing network resources to construct a next-generation Internet, thereby implementing smooth transition and avoiding excessive investment. The current Internet is based on IPv4 and cannot be transformed to the IPv6 network in a short time. Therefore, the IPv4 and IPv6 networks will coexist in a rather long time.

The coexistence, however, causes the following problems: how to keep current network services and functions at minimum cost; and how to implement transparent transmission between the IPv6 network and the IPv4 network. Network Address Translation 64 (NAT64, also called the IPv6-to-IPv4 address mapping), includes Stateful NAT64 and Stateless NAT64. Stateful NAT64 is mainly used when IPv6 network users initiate access requests to hosts/servers on the IPv4 network.

Basic Concepts

Stateful NAT64: Stateful IPv6-to-IPv4 network address translation protocol

Port Address Translation (PAT)

Network-Specific Prefix (NSP): Mainly used to check IPv6 destination addresses and IPv6 network addresses mapping to IPv4 host addresses.

Well-Known Prefix (WKP): Network prefix used by Stateful NAT64. It is used by default with the value of 64:ff9b::/96.

Working Principle

Stateful NAT64 provides a translation mechanism between IPv6 packets and IPv4 packets. This mechanism uses a Stateful NAT6 IPv6 prefix to implement translation from IPv4 host addresses to IPv6 addresses and takes NAT to implement translation from IPv6 host addresses to IPv4 addresses. Moreover, Statefull NAT64 performs protocol translation. NAT64 implements intercommunication between the pure IPv6 network and the IPv4 network.

Protocol Specification

RFC6052: IPv6 Addressing of IPv4/IPv6 Translators

RFC6144: Framework for IPv4/IPv6 Translation

RFC6145: IP/ICMP Translation Algorithm

RFC6146: Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers

Typical Application

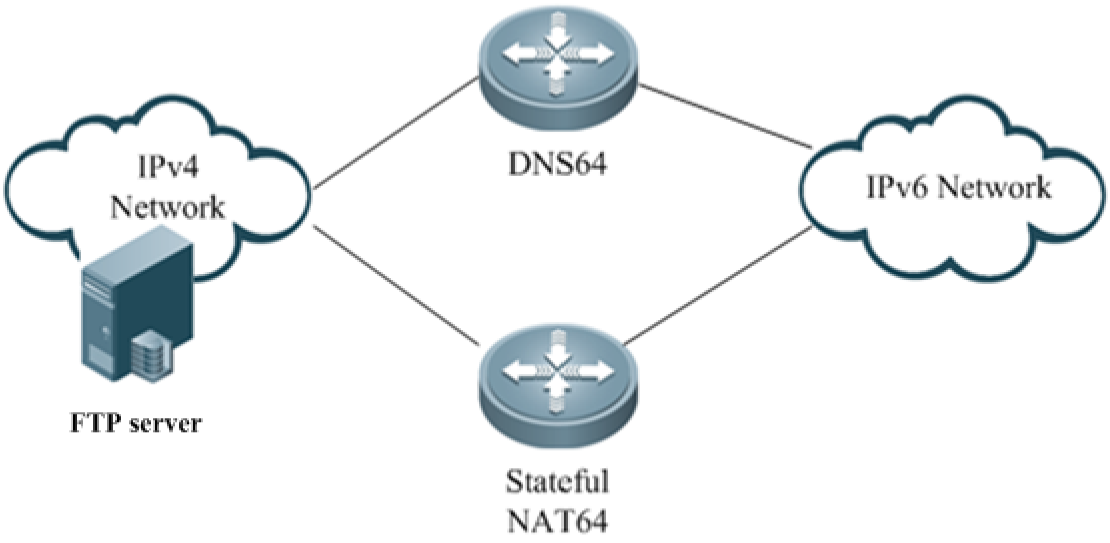


Figure 1-1 IPv6 Network Initiating a Session to IPv4 Network

Configuring Stateful NAT64

Default Configuration

Feature	Default setting
Stateful NAT64	Disabled
Default WKP	64:ff9b::/96

Configuring Static NAT64

Command	Function
Ruijie>**enable**	Enters privileged EXEC mode.
Ruijie#**configure terminal**	Enters global configuration mode.
Ruijie(config)#**ipv6 unicast-routing**	(Optional) Enables unicast routing, which is enabled by default.
Ruijie(config)#**interface** *interface-name interface-number*	Specifies an IPv6 network interface and enters interface configuration mode.
Ruijie(config-if)#**ipv6 enable**	Enables IPv6.
Ruijie(config-if)# **ipv6 address** *ipv6-address/prefix-length*	Configures the IPv6 address of the interface.
Ruijie(config-if)#**nat64 enable**	Enables NAT64 on the interface.
Ruijie(config-if)#**exit**	Exits interface configuration mode and returns to global configuration mode.
Ruijie(config)#**interface** *interface-name interface-number*	Specifies an IPv4 network interface and enters interface configuration mode.
Ruijie(config-if)#**ip address** *ip-address mask*	Configures the IPv4 address of the interface.
Ruijie(config-if)#**nat64 enable**	Enables NAT64 on the interface.

Ruijie(config-if)#**exit**	Exits interface configuration mode and returns to global configuration mode.
Ruijie(config)#**nat64 prefix stateful** *ipv6-address/prefix-length*	Allocates an IPv6 prefix as a global Stateful NAT64 prefix.
Ruijie(config)#**nat64 v6v4 static** *ipv6-address ipv4-address*	Configures NAT64 to map an IPv6 address to an IPv4 address in static mode.
Ruijie(config)#**end**	Exits global configuration mode and retunes to privileged mod.

The following example configures static Stateful NAT64.

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)#interface gigabitethernet 0/0/0
Ruijie(config-if)#ipv6 enable
Ruijie(config-if)#ipv6 address 2001:db8:1::1/96
Ruijie(config-if)#nat64 enable
Ruijie(config-if)#exit
Ruijie(config)#interface gigabitethernet 1/2/0
Ruijie(config-if)#ip address 209.165.201.1 255.255.255.0
Ruijie(config-if)#nat64 enable
Ruijie(config-if)#exit
Ruijie(config)#nat64 prefix stateful 2001:db8:0:1::/96
Ruijie(config)#nat64 v6v4 static 2001:db8:1::fffe 209.165.201.2
Ruijie(config)#end
```

Configuring Dynamic NAT64

Command	Function
Ruijie>**enable**	Enters privileged EXEC mode.
Ruijie**#configure terminal**	Enters global configuration mode.
Ruijie(config)#**ipv6 unicast-routing**	Enables unicast routing, which is enabled by default.
Ruijie(config)#**interface** *interface-name interface-number*	Specifies an IPv6 network interface and enters interface configuration mode.
Ruijie(config-if)#**ipv6 enable**	Enables IPv6.
Ruijie(config-if)# **ipv6 address** *ipv6-address/prefix-length*	Configures the IPv6 address of the interface.
Ruijie(config-if)#**nat64 enable**	Enables NAT64 on the interface.
Ruijie(config-if)#**exit**	Exits interface configuration mode and returns to global configuration mode.
Ruijie(config)#**interface** *interface-name interface-number*	Specifies an IPv4 network interface and enters interface configuration mode.
Ruijie(config-if)#**ip address** *ip-address mask*	Configures the IPv4 address of the interface.
Ruijie(config-if)#**nat64 enable**	Enables NAT64 on the interface.
Ruijie(config-if)#**exit**	Exits interface configuration mode and returns to global configuration mode.

Ruijie(config)#**ipv6 access-list** *access-list-name*	Configures an IPv6 ACL permit entry and enters IPv6 ACL mode.
Ruijie(config-ipv6-acl)#**permit ipv6** *ipv6-address* **any**	Filters IPv6 addresses based on ACL.
Ruijie(config-ipv6-acl)#**exit**	Exits interface configuration mode and returns to global configuration mode.
Ruijie(config)#**nat64 prefix stateful** *ipv6-address/prefix-length*	Allocates an IPv6 prefix as a global Stateful NAT64 prefix.
Ruijie(config)#**nat64 v4 pool** *pool-name start-ip-address end-ip-address*	Configures a NAT64 IPv4 address pool.
Ruijie(config)#**nat64 v6v4 list** *access-list-name* **pool** *pool-name*	Enables dynamic **NAT64.** *access-list-name* specifies an IPv6 ACL. The IPv6 addresses match the ACL are mapped to the addresses in *pool-name* address pool. *pool-name* specifies the name of an IPv4 address pool.
Ruijie(config)#**end**	Exits global configuration mode and returns to privileged EXEC mode.

The following example configures dynamic Stateful NAT64.

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)#interface gigabitethernet 0/0/0
Ruijie(config-if)#ipv6 enable
Ruijie(config-if)#ipv6 address 2001:db8:1::1/96
Ruijie(config-if)#nat64 enable
Ruijie(config-if)#exit
Ruijie(config)#interface gigabitethernet 0/0/1
Ruijie(config-if)#ip address 209.165.201.24 255.255.255.0
Ruijie(config-if)#nat64 enable
Ruijie(config-if)#exit
Ruijie(config)#ipv6 access-list nat64-acl
Ruijie(config-ipv6-acl)#permit ipv6 2001:db8:2::/96 any
Ruijie(config-ipv6-acl)#exit
Ruijie(config)#nat64 prefix stateful 2001:db8:1::/96
Ruijie(config)#nat64 v4 pool v4pool 209.165.201.1 209.165.201.254
Ruijie(config)#nat64 v6v4 list nat64-acl pool v4pool
Ruijie(config)#end
```

Configuring Dynamic PAT-Based NAT64

Command	Function
Ruijie>**enable**	Enters privileged EXEC mode.
Ruijie#**configure terminal**	Enters global configuration mode.
Ruijie(config)#**ipv6 unicast-routing**	(Optional) Enables unicast routing, which is enabled by default.

Ruijie(config)#**interface** *interface-name interface-number*	Specifies an IPv6 network interface and enters interface configuration mode.
Ruijie(config-if)#**ipv6 enable**	Enables IPv6.
Ruijie(config-if)# **ipv6 address** *ipv6-address/prefix-length*	Configures the IPv6 address of the interface.
Ruijie(config-if)#**nat64 enable**	Enables NAT64 on the interface.
Ruijie(config-if)#**exit**	Exits interface configuration mode and returns to global configuration mode.
Ruijie(config)#**interface** *interface-name interface-number*	Specifies an IPv4 network interface and enters interface configuration mode.
Ruijie(config-if)#**ip address** *ip-address mask*	Configures the IPv4 address of the interface.
Ruijie(config-if)#**nat64 enable**	Enables NAT64 on the interface.
Ruijie(config-if)#**exit**	Exits interface configuration mode and returns to global configuration mode.
Ruijie(config)#**ipv6 access-list** *access-list-name*	Configures an IPv6 ACL permit entry and enters IPv6 ACL mode.
Ruijie(config-ipv6-acl)#**permit ipv6** *ipv6-address* **any**	Filters IPv6 addresses based on ACL.
Ruijie(config-ipv6-acl)#**exit**	Exits interface configuration mode and returns to global configuration mode.
Ruijie(config)#**nat64 prefix stateful** *ipv6-address/prefix-length*	Allocates an IPv6 prefix as a global Stateful NAT64 prefix.
Ruijie(config)#**nat64 v4 pool** *pool-name start-ip-address end-ip-address*	Configures a NAT64 IPv4 address pool.
Ruijie(config)#**nat64 v6v4 list** *access-list-name* **pool** *pool-name* **overload**	Enables dynamic NAT64. *access-list-name* specifies an IPv6 ACL. The IPv6 addresses match the ACL are mapped to the addresses in *pool-name* address pool. *pool-name* specifies the name of an IPv4 address pool.
Ruijie(config)#**end**	Exits global configuration mode and returns to privileged EXEC mode.

The following example configures dynamic Stateful NAT64.

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)#interface gigabitethernet 0/0/0
Ruijie(config-if)#ipv6 enable
Ruijie(config-if)#ipv6 address 2001:db8:1::1/96
Ruijie(config-if)#nat64 enable
Ruijie(config-if)#exit
Ruijie(config)#interface gigabitethernet 0/0/1
Ruijie(config-if)#ip address 209.165.201.24 255.255.255.0
Ruijie(config-if)#nat64 enable
Ruijie(config-if)#exit
Ruijie(config)#ipv6 access-list nat64-acl
Ruijie(config-ipv6-acl)#permit ipv6 2001:db8:2::/96 any
Ruijie(config-ipv6-acl)#exit
Ruijie(config)#nat64 prefix stateful 2001:db8:0:1::/96
Ruijie(config)#nat64 v4 pool v4pool 209.165.201.1 209.165.201.254
Ruijie(config)#nat64 v6v4 list nat64-acl pool v4pool overload
Ruijie(config)#end
```

Configuring VRF-Based Stateful NAT64

Command	Function
Ruijie>**enable**	Enters privileged EXEC mode.
Ruijie**#configure terminal**	Enters global configuration mode.
Ruijie(config)#**ipv6 unicast-routing**	(Optional) Enables unicast routing, which is enabled by default.
Ruijie(config)#**vrf definition** *vrf-name*	Creates a multi-protocol VRF.
Ruijie(config-vrf)#**address-family ipv4**	Enables an IPv4 address family.
Ruijie(config-vrf-af)#**exit-address-family**	Exits a VRF address family.
Ruijie(config-vrf)#**address-family ipv6**	Enables an IPv6 address family.
Ruijie(config-vrf-af)#**exit-address-family**	Exits the VRF address family.
Ruijie(config-vrf)#**interface** *interface-name interface-number*	Specifies an IPv6 network interface and enters interface configuration mode.
Ruijie(config-if)#**vrf forwarding** *vrf-name*	Binds the multi-protocol VRF on the interface.
Ruijie(config-if)#**ipv6 enable**	Enables IPv6.
Ruijie(config-if)# **ipv6 address** *ipv6-address/prefix-length*	Configures the IPv6 address of the interface.
Ruijie(config-if)#**nat64 enable**	Enables NAT64 on the interface.
Ruijie(config-if)#**exit**	Exits interface configuration mode and returns to global configuration mode.
Ruijie(config)#**interface** *interface-name interface-number*	Specifies an IPv4 network interface and enters interface configuration mode.
Ruijie(config)#**vrf forwarding** *vrf-name*	Binds the multi-protocol VRF on the interface. The IPv4 and IPv6 protocol families of the multi-protocol VRF need to be enabled.
Ruijie(config-if)#**ip address** *ip-address mask*	Configures the IPv4 address of the interface.
Ruijie(config-if)#**nat64 enable**	Enables NAT64 on the interface.
Ruijie(config-if)#**exit**	Exits interface configuration mode and returns to global configuration mode.
Ruijie(config)#**ipv6 access-list** *access-list-name*	Configures an IPv6 ACL permit entry and enters IPv6 ACL mode.
Ruijie(config-ipv6-acl)#**permit ipv6** *ipv6-address* **any**	Filters IPv6 addresses based on ACL.
Ruijie(config-ipv6-acl)#**exit**	Exits interface configuration mode and returns to global configuration mode.
Ruijie(config)#**nat64 prefix stateful** *ipv6-address/prefix-length* [**vrf** *vrf-name*]	Allocates an IPv6 prefix to *vrf-name* as a Stateful NAT64 prefix.
Ruijie(config)#**nat64 v4 pool** *pool-name start-ip-address end-ip-address* [**vrf** *vrf-name*]	Configures a NAT64 IPv4 address pool for *vrf-name*.

Ruijie(config)#**nat64 v6v4 list** *access-list-name* **pool** *pool-name* [**vrf** *vrf-name*]	Enables dynamic NAT64. *access-list-name* specifies an IPv6 ACL. The IPv6 addresses match the ACL are mapped to the addresses in *pool-name* address pool. *pool-name* specifies the name of an IPv4 address pool. *vrf-name* specifies the name of the VRF.
Ruijie(config)#**ipv6 route vrf** *vrf-name ipv6-prefix/prefix-length interface next-hop*	Adds an IPv6 route. *vrf-name* specifies the name of the VRF; *ipv6-prefix* specifies an IPv6 prefix; *prefix-length* specifies the length of the IPv6 prefix; *interface* specifies an outbound interface; *next-hop* specifies a next-hop address.
Ruijie(config)#**ip route vrf** *vrf-name network mask interface next-hop*	Adds an IPv4 route. *vrf-name* specifies the name of VRF; *network* specifies a destination network segment; *mask* specifies a mask; *interface* specifies an outbound interface; *next-hop* specifies a next-hop address.
Ruijie(config)#**end**	Exits global configuration mode and returns to privileged EXEC mode.

The following example configures VRF-based dynamic Stateful NAT64.

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)#vrf definition vrf-name1
Ruijie(config-vrf)#address-family ipv4
Ruijie(config-vrf-af)#exit-address-family
Ruijie(config-vrf)#address-family ipv6
Ruijie(config-vrf-af)#exit-address-family
Ruijie(config-vrf)#interface gigabitethernet 0/0/0
Ruijie(config-if)#vrf forwarding vrf-name1
Ruijie(config-if)#ipv6 enable
Ruijie(config-if)#ipv6 address 2001:db8:1::1/96
Ruijie(config-if)#nat64 enable
Ruijie(config-if)#exit
Ruijie(config)#interface gigabitethernet 0/0/1
Ruijie(config-if)#vrf forwarding vrf-name1
Ruijie(config-if)#ip address 209.165.201.24 255.255.255.0
Ruijie(config-if)#nat64 enable
Ruijie(config-if)#exit
Ruijie(config)#ipv6 access-list nat64-acl
Ruijie(config-ipv6-acl)#permit ipv6 2001:db8:2::/96 any
Ruijie(config-ipv6-acl)#exit
Ruijie(config)#nat64 prefix stateful 2001:db8:0:1::/96 vrf vrf-name1
Ruijie(config)#nat64 v4 pool v4pool 209.165.201.1 209.165.201.254 vrf vrf-name1
Ruijie(config)#nat64 v6v4 list nat64-acl pool v4pool vrf vrf-name1
Ruijie(config)#ip route vrf vrf-name1 209.165.201.0 255.255.255.0 gigabitethernet 0/0/1
Ruijie(config)#ipv6 route vrf vrf-name1 2001:db8:0:1::D1A5:C918/120 gigabitethernet 0/0/0
Ruijie(config)#end
```

Monitoring and Maintaining Stateful NAT64

Command	Function
clear nat64 stateful statistics	Clears statistics about Stateful NAT64.
debug nat64 stateful {alg | control | event | memory | packet | pool | rule | translations}	Enables NAT64 debugging. Use the **no** form of this command to disable NAT64 debugging.
show nat64 stateful debug-buf	Displays the debugging buffer.
show nat64 mappings dynamic	Displays NAT64 dynamic mapping information.
show nat64 mappings static	Displays NAT64 static mapping information.
show nat64 prefix stateful [interfaces]	Displays all configured IPv6 prefixes of Stateful NAT64.
show nat64 services	Displays related NAT64 application layer gateway (ALG) information.
show nat64 stateful statistics	Displays statistics about Stateful NAT64.
show nat64 translations	Displays translation records of NAT64.

Configuration Examples

Static NAT64 Configuration Example

Networking Requirements

Host B on the IPv6 network can initiate a session to Host A on the IPv4 network and record the session entry. To meet the requirements, deploy a NAT64 device between the IPv6 network and the IPv4 network.

Networking Topology

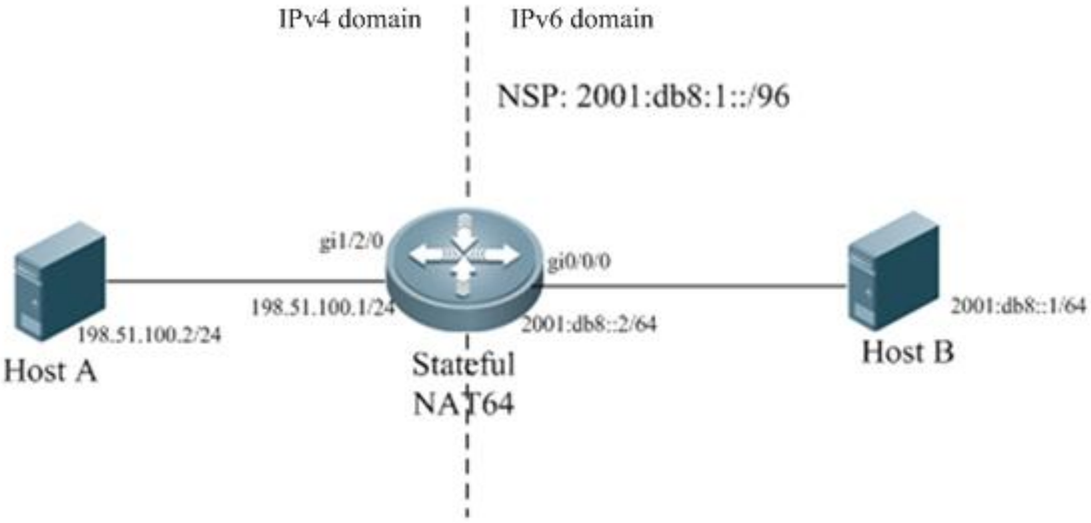


Figure 1-2 Topology of Static Stateful NAT64

Configuration Tips

- Configure the IPv6 address of an IPv6 network interface and enable NAT64 on the interface.
- Configure the IPv4 address of an IPv4 network interface and enable NAT64 on the interface.
- Configure a global NAT64 prefix.
- Configures static IPv6-to-IPv4 address translation.

Configuration Steps

1) Perform the following configurations on the Stateful NAT64 device.

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)#interface gigabitethernet 0/0/0
Ruijie(config-if)#ipv6 enable
Ruijie(config-if)#ipv6 address 2001:db8::2/64
Ruijie(config-if)#nat64 enable
Ruijie(config-if)#exit
Ruijie(config)#interface gigabitethernet 1/2/0
Ruijie(config-if)#ip address 198.51.100.2 255.255.255.0
Ruijie(config-if)#nat64 enable
Ruijie(config-if)#exit
Ruijie(config)#nat64 prefix stateful 2001:db8:1::/96
Ruijie(config)#nat64 v6v4 static 2001:db8::1 198.51.100.4
Ruijie(config)#end
```

2) On Host B

Configure the IPv6 address 2001:db8::1/64 on Host B and configure a static route to the prefix 2001:db8:0:1::/96.

3) On Host A

Configure the IP address 198.51.100.2/24 on Host A.

Verification

Run the **ping 2001:db8:1::c633:6401** command on Host B.

```
Ping statistics for 2001:db8:1::c633:6401:
   Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
Ruijie#show nat64 translations
Prot  IPv4 source           IPv6 source
     IPv4 destination        IPv6 destination
icmp  198.51.100.4,47       2001:db8::1 ,47
     198.51.100.1 ,47       2001:db8:1::c633:6401,47
```

Dynamic NAT64 Configuration Example

Networking Requirements

Host B, Host C or another host on the IPv6 network can initiate a session to Host A on the IPv4 network and record the session entry. To meet the requirements, deploy a NAT64 device between the IPv6 network and the IPv4 network. Dynamic NAT64 takes effect as long as the number of IPv6-domain hosts that initiate a session to Host A does not exceed the number of IPv4 addresses in the address pool.

Networking Topology

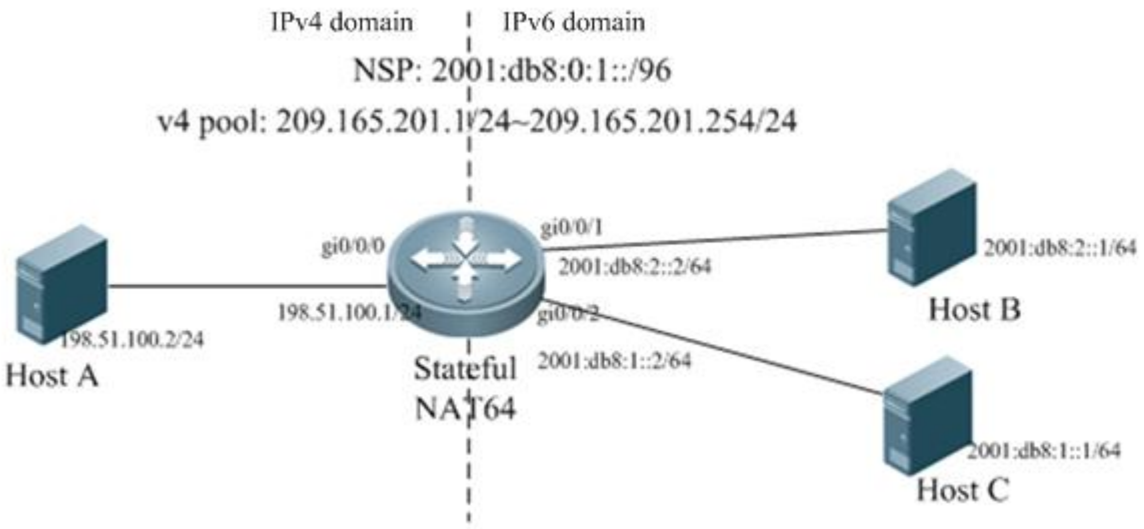


Figure 1-3 Topology of Dynamic Stateful NAT64

Configuration Tips

- Configure the IPv6 address of an IPv6 network interface and enable NAT64 on the interface.
- Configure the IPv4 address of an IPv4 network interface and enable NAT64 on the interface.
- Configure an ACL entry.
- Configure a global NAT64 prefix.
- Configures an IPv4 address pool.
- Configure a dynamic IPv6-to-IPv4 address translation list.

Configuration Steps

1) Perform the following configurations on the Stateful NAT64 device.

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)#interface gigabitethernet 0/0/1
Ruijie(config-if)#ipv6 enable
Ruijie(config-if)#ipv6 address 2001:db8:2::2/64
Ruijie(config-if)#nat64 enable
Ruijie(config-if)#exit
Ruijie(config)#interface gigabitethernet 0/0/2
Ruijie(config-if)#ipv6 enable
Ruijie(config-if)#ipv6 address 2001db8:1::2/64
Ruijie(config-if)#nat64 enable
Ruijie(config-if)#exit
Ruijie(config)#interface gigabitethernet 0/0/0
Ruijie(config-if)#ip address 198.51.100.1 255.255.255.0
Ruijie(config-if)#nat64 enable
Ruijie(config-if)#exit
Ruijie(config)#ipv6 access-list v6_list1
Ruijie(config-ipv6-acl)#permit ipv6 any any
Ruijie(config-ipv6-acl)#exit
Ruijie(config)#nat64 prefix stateful 2001:db8:0:1::/96
Ruijie(config)#nat64 v4 pool v4_pool 209.165.201.1 209.165.201.254
Ruijie(config)#nat64 v6v4 list v6_list1 pool v4_pool
Ruijie(config)#end
```

2) On Host B

Configure the IPv6 address 2001:db8:2::1/64 on Host B and configure a static route to the prefix 2001:db8:0:1::/96.

3) On Host C

Configure the IPv6 address 2001:db8:1::1/64 on Host C and configure a static route to the prefix 2001:db8:0:1::/96.

4) On Host A

Configure the IP address 198.51.100.2/24 on Host A and configure a static route to the destination network segment 209.165.201.0/24.

Verification

Run the **ping 2001:db8:0:1::c633:6401** command on Host B.

```
Ping statistics for 2001:db8:0:1::c633:6401:
   Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Run the **ping 2001:db8:0:1::c633:6401** command on Host C.

```
Ping statistics for 2001:db8:0:1::c633:6401:
   Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
Ruijie#show nat64 translations
Prot  IPv4 source           IPv6 source
     IPv4 destination        IPv6 destination
---   ----                 -----
```

```
icmp  209.165.201.1,47        2001:db8:2::1,47
     198.51.100.1 ,47       2001:db8:0:1::c633:6401,47
icmp  209.165.201.2,47        2001:db8:1::1,47
      198.51.100.1,47       2001:db8:0:1::c633:6401,47
```

Configuration Example of Dynamic PAT-Based NAT64

Networking Requirements

When the number of IPv6-domain hosts that initiate a session to Host A does not exceed the number of IPv4 addresses in the address pool, Host B or Host C on the IPv6 network can initiate a session to Host A on the IPv4 network and record the session entry. Otherwise, deploy a dynamic PAT-based NAT64 device between the IPv6 network and the IPv4 network, so that Host B or Host C can continue to access Host A.

Networking Topology

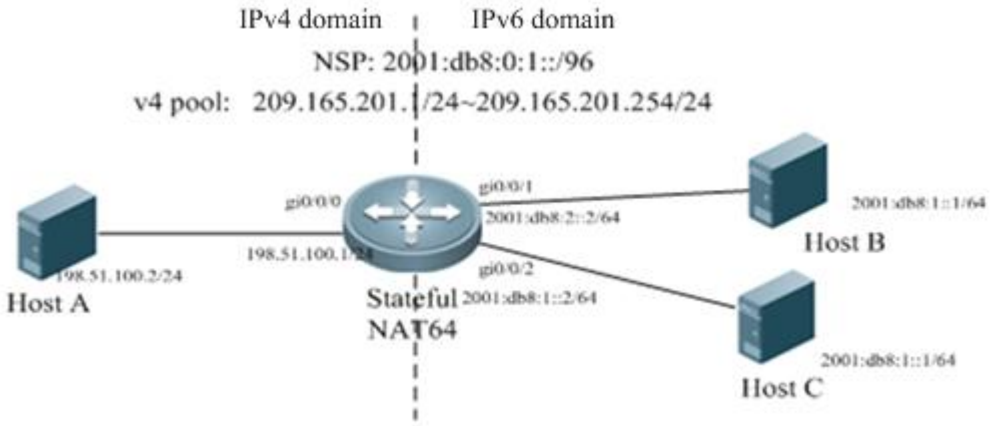


Figure 1-4 Topology of Dynamic PAT-Based NAT64

Configuration Tips

- Configure the IPv6 address of an IPv6 network interface and enable NAT64 on the interface.
- Configure the IPv4 address of an IPv4 network interface and enable NAT64 on the interface.
- Configure an ACL entry.
- Configure a global NAT64 prefix.
- Configures an IPv4 address pool.
- Configure a dynamic PAT-based IPv6-to-IPv4 address translation list.

Configuration Steps

1) Perform the following configurations on the Stateful NAT64 device.

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)#interface gigabitethernet 0/0/1
Ruijie(config-if)#ipv6 enable
Ruijie(config-if)#ipv6 address 2001:db8:2::2/64
Ruijie(config-if)#nat64 enable
Ruijie(config-if)#exit
Ruijie(config)#interface gigabitethernet 0/0/2
Ruijie(config-if)#ipv6 enable
Ruijie(config-if)#ipv6 address 2001:db8:1::2/64
Ruijie(config-if)#nat64 enable
Ruijie(config-if)#exit
Ruijie(config)#interface gigabitethernet 0/0/0
Ruijie(config-if)#ip address 198.51.100.1 255.255.255.0
Ruijie(config-if)#nat64 enable
Ruijie(config-if)#exit
Ruijie(config)#ipv6 access-list v6_list1
Ruijie(config-ipv6-acl)#permit ipv6 any any
Ruijie(config-ipv6-acl)#exit
Ruijie(config)#nat64 prefix stateful 2001:db8:0:1::/96
Ruijie(config)#nat64 v4 pool v4_pool 209.165.201.1 209.165.201.254
Ruijie(config)#nat64 v6v4 list v6_list1 pool v4_pool overload
Ruijie(config)#end
```

2) On Host B

Configure the IPv6 address 2001:db8:2::1/64 on Host B and configure a static route to the prefix 2001:db8:0:1::/96.

3) On Host C

Configure the IPv6 address 2001:db8:1::1/64 on Host B and configure a static route to the prefix 2001:db8:0:1::/96.

4) On Host A

Configure the IPv address 198.51.100.2/24 on Host A and configure a static route to the destination network segment 209.165.201.0/24.

Verification

Run the **ping 2001:db8:0:1::c633:6401** command on Host B.

```
Ping statistics for 2001:db8:0:1::c633:6401:
    Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Run the **ping 2001:db8:0:1::c633:6401** command on Host C.

```
Ping statistics for 2001:db8:0:1::c633:6401:
    Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
Ruijie#show nat64 translations
```

```
Prot  IPv4 source             IPv6 source
     IPv4 destination         IPv6 destination
---   ----                    -----
icmp  209.165.201.1,47         2001:db8:2::1 ,47
      198.51.100.1,47        2001:db8:0:1::c633:6401,47
icmp  209.165.201.1,1029        2001:db8:1::1, 1029
      198.51.100.1, 1029      2001:db8:0:1::c633:6401, 1029
```

Configuration Example of VRF-Based Stateful NAT64

Networking Requirements

Only hosts on the IPv6 network can initiate a session to Host A on the IPv4 network. Meanwhile, the router device can be divided into independently logical routers. To meet the requirements, deploy a VRF-based Stateful NAT64 device on the boundary between the IPv6 network and the IPv4 network, so that logical translation devices can communicate with each other.

Networking Topology

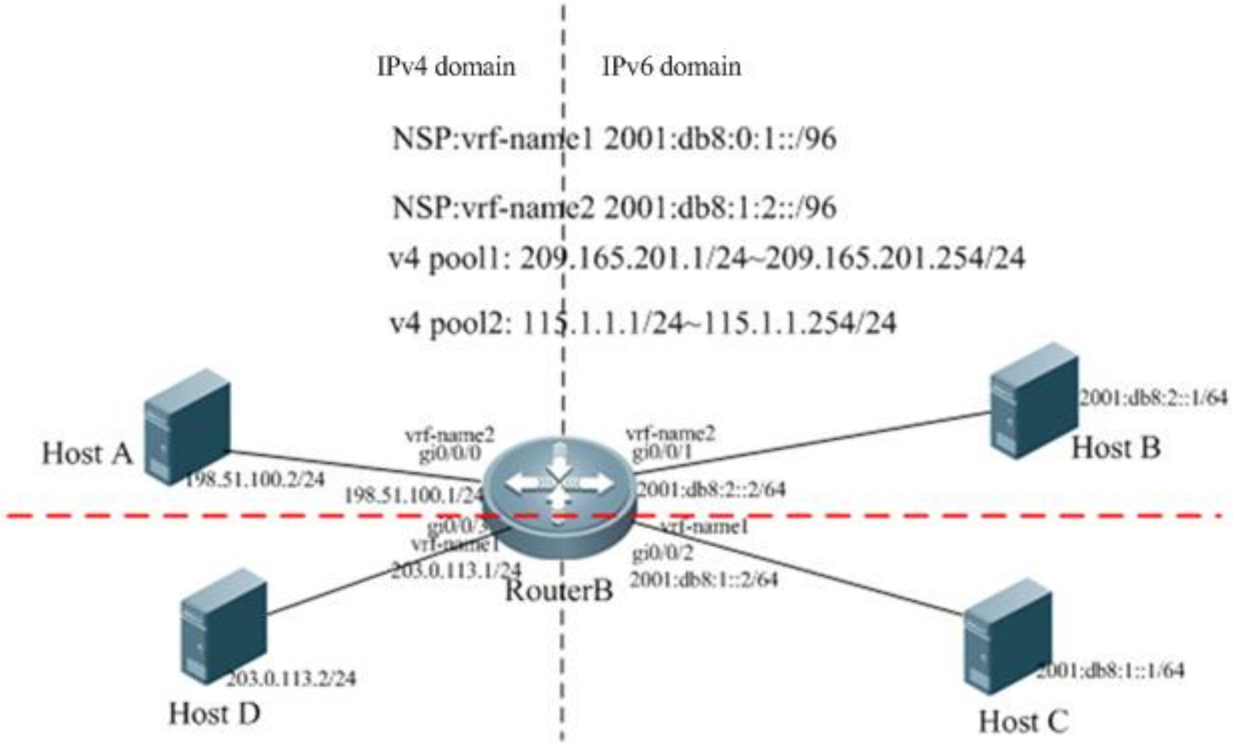


Figure 1-5 Topology of VRF-Based Stateful NAT64

Configuration Tips

- Create VRFs.
- Enable VRFs on interfaces.
- Configure the IPv6 address of an IPv6 network interface and enable NAT64 on the interface.

- Configure the IPv4 address of an IPv4 network interface and enable NAT64 on the interface.
- Configure a routing protocol.
- Configure an ACL.
- Configure global NAT64 prefixes for different VRFs.
- Configure IPv4 address pools for different VRFs.
- Configure dynamic IPv6-to-IPv4 translation access lists for different VRFs.

Configuration Steps

1) Perform the following configurations on Router B, which serves as the Stateful NAT64 device.

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)#vrf definition vrf-name1
Ruijie(config-vrf)#address-family ipv4
Ruijie(config-vrf-af)#exit-address-family
Ruijie(config-vrf)#address-family ipv6
Ruijie(config-vrf-af)#exit-address-family
Ruijie(config-vrf)#interface gigabitethernet 0/0/0
Ruijie(config-if)#vrf forwarding vrf-name1
Ruijie(config-if)#ip address 198.51.100.1 255.255.255.0
Ruijie(config-if)#nat64 enable
Ruijie(config-if)#exit
Ruijie(config)#interface gigabitethernet 0/0/1
Ruijie(config-if)#vrf forwarding vrf-name1
Ruijie(config)#ipv6 enable
Ruijie(config-if)#ipv6 address 2001:db8:2::2/64
Ruijie(config-if)#nat64 enable
Ruijie(config-if)#exit
Ruijie(config)#vrf definition vrf-name2
Ruijie(config-vrf)#address-family ipv4
Ruijie(config-vrf-af)#exit-address-family
Ruijie(config-vrf)#address-family ipv6
Ruijie(config-vrf-af)#exit-address-family
Ruijie(config-vrf)#interface gigabitethernet 0/0/3
Ruijie(config-if)#vrf forwarding vrf-name2
Ruijie(config-if)#ip address 203.0.113.2 255.255.255.0
Ruijie(config-if)#nat64 enable
Ruijie(config-if)#exit
Ruijie(config-vrf)#interface gigabitethernet 0/0/2
Ruijie(config-if)#vrf forwarding vrf-name2
Ruijie(config-if)#ipv6 address 2001:db8:1::2/64
Ruijie(config-if)#nat64 enable
Ruijie(config-if)#exit
Ruijie(config)#ipv6 access-list nat64-acl1
Ruijie(config-ipv6-acl)#permit ipv6 2001:db8:2::/64 any
Ruijie(config-ipv6-acl)#exit
Ruijie(config)#nat64 prefix stateful 2001:db8:0:1::/96 vrf vrf-name1
Ruijie(config)#nat64 v4 pool v4pool1 209.165.201.1 209.165.201.254
Ruijie(config)#nat64 v4 pool v4pool2 115.1.1.1 115.1.1.20
Ruijie(config)#nat64 v6v4 list nat64-acl1 pool v4pool1 vrf vrf-name1
Ruijie(config)#ipv6 access-list nat64-acl2
Ruijie(config-ipv6-acl)#permit ipv6 4001::/64 any
Ruijie(config-ipv6-acl)#exit
Ruijie(config)#nat64 prefix stateful 2001:db8:1:2::/96 vrf vrf-name2
Ruijie(config)#nat64 v6v4 list nat64-acl2 pool v4pool2 vrf vrf-name2
Ruijie(config)#ip route vrf vrf-name1 209.165.201.0 255.255.255.0 gi0/0/0
Ruijie(config)#ip route vrf vrf-name1 209.165.201.0 255.255.255.0 gi0/0/3
Ruijie(config)#ipv6 route vrf vrf-name1 2001:db8:0:1::/96 gi0/0/2
Ruijie(config)#ipv6 route vrf vrf-name1 2001:db8:0:1::/96 gi0/0/1
Ruijie(config)#ipv6 route vrf vrf-name2 2001:db8:1:2::/96 gi0/0/1
Ruijie(config)#ipv6 route vrf vrf-name2 2001:db8:1:2::/96 gi0/0/2
Ruijie(config)#ip route vrf vrf-name2 115.1.1.0 255.255.255.0 gi0/0/0
Ruijie(config)#ip route vrf vrf-name2 115.1.1.0 255.255.255.0 gi0/0/3
Ruijie(config)#end
```

2) On Host B

Configure the IPv6 address 2001:db8:2::1/64 on Host B and configure a static route to the prefix 2001:db8:0:1::/96.

3) On Host C

Configure the IPv6 address 2001:db8:1::1/64 on Host C and configure static routes to the prefixes 2001:db8:0:1::/96 and 2001:db8:1:2::/96.

4) On Host A

Configure the IP address 198.51.100.2/24 on Host A and configure a static route to the destination network segment 209.165.201.0/24.

5) On Host D

Configure the IPv6 address 203.0.113.2/24 on Host D and configure a static route to the destination network segment 209.165.201.0/24.

Verification

Run the **ping 2001:db8:0:1::c633:6401** command on Host B.

```
Ping statistics for 2001:db8:0:1::c633:6401:
   Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Run the **ping 2001:db8:0:1::c633:6401** command on Host C.

```
Ping statistics for 2001:db8:0:1::c633:6401:
   Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
Ruijie#show nat64 translations
Prot  IPv4 source            IPv6 source
     IPv4 destination         IPv6 destination
---   ----                   -----
icmp  209.165.201.1,47        2001:db8:2::1 ,47
      198.51.100.1,47       2001:db8:0:1::c633:6401,47
icmp  209.165.201.2,1027      2001:db8:1::1 ,1027
     198.51.100.1 ,1027     2001:db8:0:1::c633:6401,1027
```

Configuration Example of ALG-Based Stateful NAT64

Networking Requirements

The various typical applications in address translation scenario on the pure IPv6 network or IPv4 network need not only address translation but also application-layer information transformation. For example, when IPv6 users on the IPv6 network initiate access requests to the IPv4 FTP server, the FTP ALG function needs to be added to the NAT64 device to meet application requirements. NAT64 can only translate addresses.

When some applications (such as DNS, VoIP, and multimedia applications) perform address family traversal, the corresponding protocols must ensure that the traversal function works well. NAT64 is only a transition technology and cannot meet all application requirements.

Networking Topology

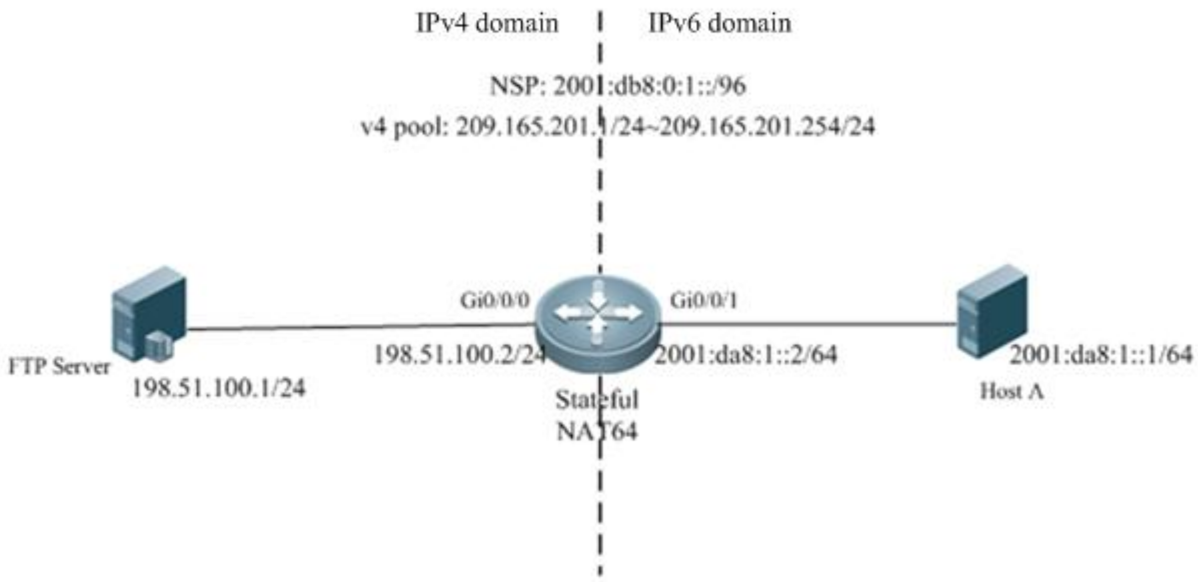


Figure 1-6 Topology of FTP ALG-Based Stateful NAT64

Configuration Tips

- Configure the IPv6 address of an IPv6 network interface and enable NAT64 on the interface.
- Configure the IPv4 address of an IPv4 network interface and enable NAT64 on the interface.
- Configure an ACL entry.
- Configure a global NAT64 prefix.
- Configures an IPv4 address pool.
- Configure a dynamic IPv6-to-IPv4 address translation list.
- Enable FTP ALG (enabled by default).

Configuration Steps

1) Perform the following configurations on the Stateful NAT64 device.

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)#interface gigabitethernet 0/0/1
Ruijie(config-if)#ipv6 enable
Ruijie(config-if)#ipv6 address 2001:da8:1::2/64
Ruijie(config-if)#nat64 enable
Ruijie(config-if)#exit
Ruijie(config)#interface gigabitethernet 0/0/0
Ruijie(config-if)#ipv4 address 198.51.100.2 255.255.255.0
Ruijie(config-if)#nat64 enable
Ruijie(config-if)#exit
Ruijie(config)#ipv6 access-list v6_list1
Ruijie(config-ipv6-acl)#permit ipv6 any any
Ruijie(config-ipv6-acl)#exit
Ruijie(config)#nat64 prefix stateful 2001:db8:0:1::/96
Ruijie(config)#nat64 v4 pool v4_pool 209.165.201.1 209.165.201.254
Ruijie(config)#nat64 v6v4 list v6_list1 pool v4_pool
Ruijie(config)#end
```

2) On Host A in the IPv6 domain

Configure the IP address 2001:da8:1::1/64 for Host A and configure a corresponding static route.

3) On the FTP server

Configure the IPv4 address as 198.51.100.1/24 and configure a static route to the destination network segment 209.165.201.0/24.

Verification

1) Run the **ping 2001:db8:0:1::c633:6401** command on Host A.

```
Ping statistics for 2001:db8:0:1::c633:6401:
   Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
```

```
Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

2) Enter FTP mode.

Running the **put**, **get**, and **dir** commands can upload a file to the FTP server or download a file from it.

Configuring Stateless NAT64

Understanding Stateless NAT64

Overview

With the fast development of the Internet, IPv4 can no longer meet Internet requirements. Under this circumstance, IPv6 is about to be deployed. To support an IPv6 network, you must make full use of existing network resources to construct a next-generation Internet, thereby implementing smooth transition and avoiding excessive investment. The current Internet is based on IPv4 and cannot be transited to the IPv6 network in a short time. Therefore, the IPv4 and IPv6 networks will coexist in a rather long time.

The coexistence however, causes the following problems: how to keep current network services and functions at minimum cost; and how to implement transparent transmission between the IPv6 network and the IPv4 network. Network Address Translation 64 (NAT64, also called the IPv6-to-IPv4 network address translation protocol) includes Stateful NAT64 and Stateless NAT64. Stateless NAT64 is mainly used when IPv4 network users initiate access requests to hosts on the IPv6 network.

Basic Concepts

Stateless Network Address Translation 64 (Stateless NAT64): Stateless IPv6-to-IPv4 network address translation protocol. Stateless NAT64 provides a translation mechanism, which implements translation between IPv4 addresses and IPv6 addresses. The translation involves parsing the entire IPv6 header, obtaining related information and translating it into an IPv4 header or a completely converse translation process. Stateless NAT64 can translate the IP addresses of only some types of ICMPv4 and ICMPv6 packets due to the protocol features, such as translation between ICMPv4 request/response packets and ICMPv6 request/response packets and translation between unreachable ICMPv4 packets and unreachable ICMPv6 packets. Address translation for each packet relies on interface configurations. Stateless NAT64 does not maintain data flow statuses.

IPv4-translatable IPv6 address: IPv6 address after Stateless NAT64 that is allocated to an IPv6 host

☑ Stateless NAT64 can be used only when there are IPv4-translatable IPv6 addresses.

☑ Stateless NAT64 does not support multicast.

☑ Stateless NAT64 cannot be used by the application without a corresponding ALG.

☑ Stateless NAT64 cannot translate an IPv4 option, an IPv6 routing header in an IPv6 extension header, a hop-by-hop extension header, or a destination option header.

Working Principle

Stateless NAT64 works on the boundary device between the IPv6 network and the IPv4 network. The Stateless NAT64 module translates IP headers between the two networks and performs semantic translation for packets according to different protocols so as to implement transparent transmission between the two networks.

Protocol Specification

RFC6052: IPv6 Addressing of IPv4/IPv6 Translators

RFC6144: Framework for IPv4/IPv6 Translation

RFC6145: IP/ICMP Translation Algorithm

Typical Application

Application Scenario 1

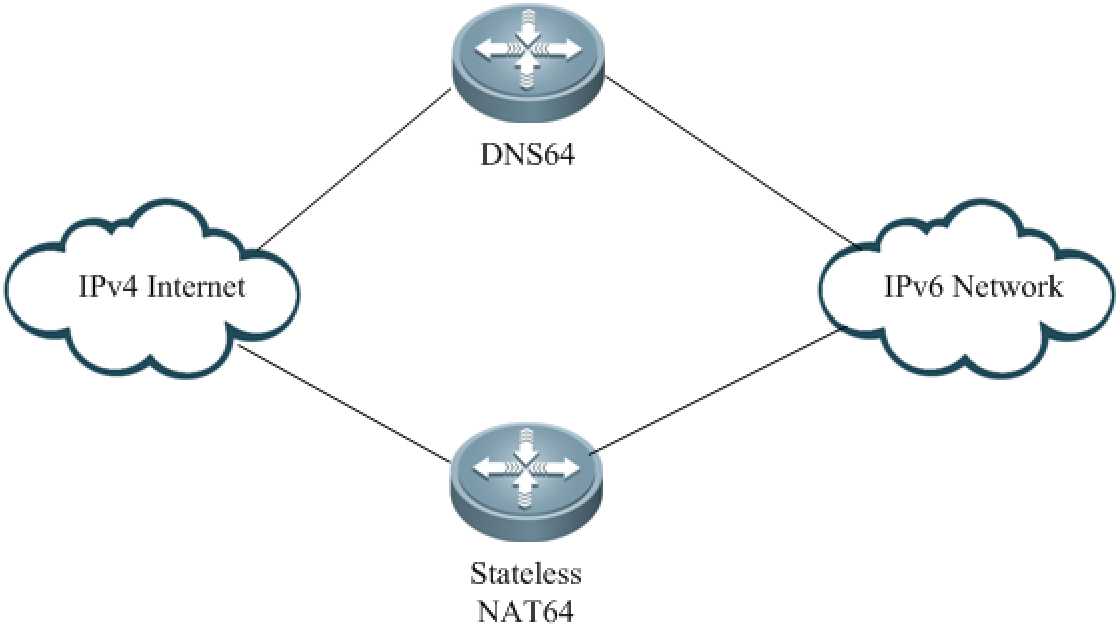


Figure 2-1 Interaction Between the IPv4 Internet and the IPv6 Network

This application scenario supports IPv4 network users to access IPv6 network resources.

Users of the IPv4 Internet can access IPv6 network resources in a specific scope.

This scenario has the following functions:

3) A new IPv6 content provider can provide resources for users both on the IPv6 network and the IPv4 Internet.

4) An IPv4 content provider that is transited to the pure IPv6 network can still provide resources for original IPv4 users and keep the connections with them.

This scenario has the following access modes:

- IP-based access
- Domain name-based access

Application Scenario 2

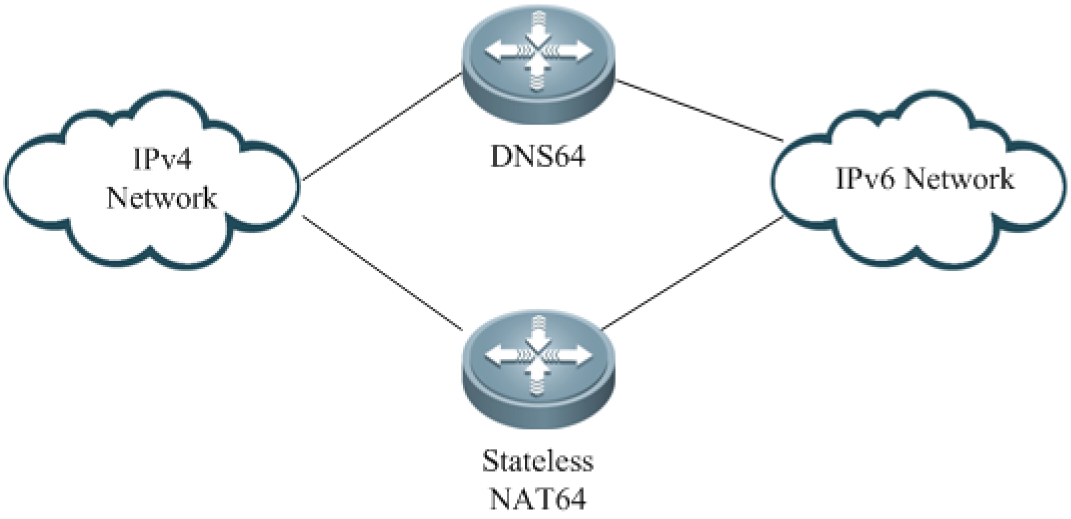


Figure 2-2 Interaction Between the IPv4 Network and the IPv6 Network

This application scenario mainly allows IPv4 network users to initiate access requests.

This scenario supports interaction between the IPv4 network and the IPv6 network, and mainly allows IPv4 network users to initiate access requests. The internal address can be a public address or a private address. The host addresses on the IPv6 network must be IPv4-translatable IPv6 addresses.

Configuring Stateless NAT64

Configuring Stateless NAT64

Command	Function
Ruijie>**enable**	Enters privileged EXEC mode.
Ruijie#**configure terminal**	Enters global configuration mode.
Ruijie(config)#**ipv6 unicast-routing**	(Optional) Enables unicast routing, which is enabled by default.
Ruijie(config)#**interface** *interface-name interface-number*	Specifies an IPv6 network interface and enters interface configuration mode.
Ruijie(config-if)#**ipv6 enable**	Enables IPv6.
Ruijie(config-if)# **ipv6 address** *ipv6-address/prefix-length*	Configures the IPv6 address of the interface.
Ruijie(config-if)#**nat64 enable**	Enables NAT64 on the interface.
Ruijie(config-if)#**exit**	Exits interface configuration mode and returns to global configuration mode.
Ruijie(config)#**interface** *interface-name interface-number*	Specifies an IPv4 network interface and enters interface configuration mode.
Ruijie(config-if)#**ip address** *ip-address mask*	Configures the IPv4 address of the interface.
Ruijie(config-if)#**nat64 enable**	Enables NAT64 on the interface.
Ruijie(config-if)#**exit**	Exits interface configuration mode and returns to global configuration mode.

Ruijie(config)#**nat64 prefix stateless** *ipv6-address/prefix-length*	Allocates an IPv6 prefix as a global Stateless NAT64 prefix.
Ruijie(config)#**nat64 route** *ipv4-prefix/length interface-name interface-number*	Configures a route which starts from an IPv4 network segment and is destined for a specific IPv6 interface. Running this command can configure a route with a specific prefix and the default route with the prefix 0.0.0.0/0 in a VRF.
Ruijie(config)#**ipv6 route** *ipv6-prefix/length interface-name interface-number nexthop-address*	Configures an IPv6 route, which is used to transmit the packets whose routes are changed to the destination IPv6 address.
Ruijie(config)#**end**	Exits global configuration mode and returns to privileged EXEC mode.

The following example configures Stateless NAT64.

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)#interface gigabitethernet 0/0
Ruijie(config-if)#ipv6 enable
Ruijie(config-if)#ipv6 address 2001:db8::1/96
Ruijie(config-if)#nat64 enable
Ruijie(config-if)#exit
Ruijie(config)#interface gigabitethernet 0/1
Ruijie(config-if)#ip address 198.51.100.1 255.255.255.0
Ruijie(config-if)#nat64 enable
Ruijie(config-if)#exit
Ruijie(config)#nat64 prefix stateless 2001:db8:0:1::/96
Ruijie(config)#nat64 route 203.0.113.0/24 gigabitethernet 0/0
Ruijie(config)#ipv6 route 2001:db8:0:1::/96 gigabitethernet 0/0 2001:db8::2
Ruijie(config)#ip route 0.0.0.0 0.0.0.0 198.51.100.2
Ruijie(config)#end
```

Configuring Multi-Prefix Stateless NAT64

Command	Function
Ruijie>**enable**	Enters privileged EXEC mode.
Ruijie**#configure terminal**	Enters global configuration mode.
Ruijie(config)#**ipv6 unicast-routing**	(Optional) Enables unicast routing, which is enabled by default.
Ruijie(config)#**interface** *interface-name interface-number*	Specifies an IPv6 network interface and enters interface configuration mode.
Ruijie(config-if)#**ipv6 enable**	Enables IPv6.
Ruijie(config-if)# **ipv6 address** *ipv6-address/prefix-length*	Configures the IPv6 address of the interface.
Ruijie(config-if)#**nat64 enable**	Enables NAT64 on the interface.

Ruijie(config-if)#**nat64 prefix stateless v6v4** *ipv6-address/prefix-length*	Configures the IPv6 prefix for Statelesss NAT64 (IPv6-to-IPv4 address translation) on an interface. The prefix must work with the global Stateless NAT64 prefix (v4v6). Otherwise, addresses cannot be translated.
Ruijie(config-if)#**exit**	Exits interface configuration mode and returns to global configuration mode.
Ruijie(config)#**interface** *interface-name interface-number*	Specifies an IPv4 network interface and enters interface configuration mode.
Ruijie(config-if)#**ip address** *ip-address mask*	Configures the IPv4 address of the interface.
Ruijie(config-if)#**nat64 enable**	Enables NAT64 on the interface.
Ruijie(config-if)#**exit**	Exits interface configuration mode and returns to global configuration mode.
Ruijie(config)#**nat64 prefix stateless v4v6** *ipv6-address/prefix-length*	Allocates an IPv6 prefix as a global Stateless NAT64 prefix. The prefix must work with the Stateless NAT64 prefix (v6v4). Otherwise, addresses cannot be translated.
Ruijie(config)#**nat64 route** *ipv4-prefix/length interface-name interface-number*	Configures a route which starts from an IPv4 network segment and is destined for a specific IPv6 interface. Running this command can configure a route with a specific prefix and the default route with the prefix 0.0.0.0/0 in a VRF.
Ruijie(config)#**ipv6 route** *ipv6-prefix/length interface-name interface-number*	Configures an IPv6 route, which is used to transmit the packets whose routes are changed to the destination IPv6 address.
Ruijie(config)#end	Exits global configuration mode and returns to privileged EXEC mode.

The following example configures multi-prefix Stateless NAT64.

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)#interface gigabitethernet 0/0
Ruijie(config-if)#ipv6 enable
Ruijie(config-if)#ipv6 address 2001:db8::1/96
Ruijie(config-if)#nat64 enable
Ruijie(config-if)#nat64 prefix stateless v6v4 2001:db8:0:1::/96
Ruijie(config-if)#exit
Ruijie(config)#interface gigabitethernet 0/1
Ruijie(config-if)#ip address 198.51.100.1 255.255.255.0
Ruijie(config-if)#nat64 enable
Ruijie(config-if)#exit
Ruijie(config)#nat64 prefix stateless v4v6 2001:db8:2::1/96
Ruijie(config)#nat64 route 203.0.113.0/24 gigabitethernet 0/0
Ruijie(config)#ipv6 route 2001:db8:0:1::/96 gigabitethernet 0/0 2001:db8::2
Ruijie(config)#ip route 0.0.0.0 0.0.0.0 198.51.100.2
Ruijie(config)#end
```

Configuring VRF-Based Stateless NAT64

Command	Function
Ruijie>**enable**	Enters privileged EXEC mode.
Ruijie**#configure terminal**	Enters global configuration mode.
Ruijie(config)#**ipv6 unicast-routing**	(Optional) Enables unicast routing, which is enabled by default.
Ruijie(config)#**interface** *interface-name interface-number*	Specifies an IPv6 network interface and enters interface configuration mode.
Ruijie(config-if)#**vrf forwarding vrf** *vrf-name*	Enables a VRF on an interface. The IPv4 and IPv6 protocol families of the VRF also need to be enabled.
Ruijie(config-if)#**ipv6 enable**	Enables IPv6.
Ruijie(config-if)# **ipv6 address** *ipv6-address/prefix-length*	Configures the IPv6 address of the interface.
Ruijie(config-if)#**nat64 enable**	Enables NAT64 on the interface.
Ruijie(config-if)#**exit**	Exits interface configuration mode and returns to global configuration mode.
Ruijie(config)#**interface** *interface-name interface-number*	Specifies an IPv4 network interface and enters interface configuration mode.
Ruijie(config-if)#**vrf forwarding vrf** *vrf-name*	Enables a VRF on an interface. The IPv4 and IPv6 protocol families of the VRF also need to be enabled.
Ruijie(config-if)#**ip address** *ip-address mask*	Configures the IPv4 address of the interface.
Ruijie(config-if)#**nat64 enable**	Enables NAT64 on the interface.
Ruijie(config-if)#**exit**	Exits interface configuration mode and returns to global configuration mode.
Ruijie(config)#**nat64 prefix stateless** *ipv6-address/prefix-length* **vrf** *vrf-name*	Configures Stateless NAT64 *ipv6-address/prefix-length* under *vrf-name*.
Ruijie(config)#**nat64 route** *ipv4-prefix/length interface-name interface-number* **vrf** *vrf-name*	Configures a route which starts from an IPv4 network segment and is destined for a specific IPv6 interface. Running this command can configure a route and the default route with the prefix 0.0.0.0/0 in a VRF.
Ruijie(config)#**ipv6 route** *ipv6-prefix/length interface-name interface-number*	Configures an IPv6 route, which is used to transmit the packets whose routes are changed to the destination IPv6 address.
Ruijie(config)#end	Exits global configuration mode and returns to privileged EXEC mode.

The following example configures VRF-based Stateless NAT64.

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)#interface gigabitethernet 0/0
Ruijie(config-if)# vrf forwarding 1
Ruijie(config-if)#ipv6 enable
Ruijie(config-if)#ipv6 address 2001:db8::1/96
Ruijie(config-if)#nat64 enable
Ruijie(config-if)#exit
Ruijie(config)#interface gigabitethernet 0/1
Ruijie(config-if)# vrf forwarding 1
Ruijie(config-if)#ip address 198.51.100.1 255.255.255.0
Ruijie(config-if)#nat64 enable
Ruijie(config-if)#exit
Ruijie(config)#nat64 prefix stateless 2001:db8:0:1::/96 vrf 1
Ruijie(config)#nat64 route 203.0.113.0/24 gigabitethernet 0/0 vrf 1
Ruijie(config)#ipv6 route vrf 1 2001:db8:0:1::/96 gigabitethernet 0/0 2001:db8::2
Ruijie(config)#ip route vrf 1 0.0.0.0 0.0.0.0 198.51.100.2
Ruijie(config)#end
```

Monitoring and Maintaining Stateless NAT64

Command	Function
clear nat64 stateless statistics	Clears statistics about Stateless NAT64.
debug nat64 stateless { **control** | **packet**]	Enables Stateless NAT64 debugging. Use the **no** form of this command to disable Stateless NAT64 debugging.
show nat64 stateless debug-buf	Displays the debugging buffer.
show nat64 prefix stateless [**interfaces**]	Displays all configured IPv6 prefixes of Stateless NAT64.
show nat64 stateless statistics	Displays statistics about Stateless NAT64.

Configuration Examples

Stateless NAT64 Configuration Example

Networking Requirements

Host A with the address 198.51.100.2/24 in the IPv4 domain can access Host B with the address 2001:db8:0:1::cb00:7101 in the IPv6 domain. To meet this requirement, deploy a NAT64 device (Router B) between the IPv4 domain and the IPv6 domain and configure a global IPv6 prefix for NAT64 on Router B to implement intercommunication between the two domains.

Networking Topology

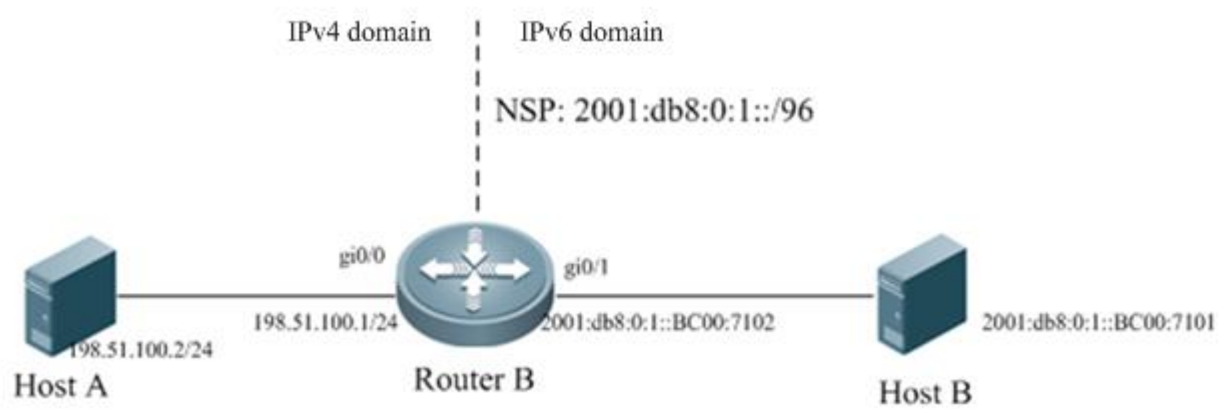


Figure 2-3 Topology of Stateless NAT64

Configuration Tips

Perform the following configurations on the Stateless NAT64 device:

- Configure the IPv6 address of an IPv6 network interface.
- Configure the IPv4 address of an IPv4 network interface.
- Configure a global NAT64 prefix and enable NAT64 on the interface.
- Configure a route which starts from the IPv4 network segment and is destined for the interface for NAT64.
- Configure a static route that is used to transmit translated packets to the IPv6 address.

Configuration Steps

5) Perform the following configurations on Router B, which serves as the Stateless NAT64 device.

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)#interface gigabitethernet 0/1
Ruijie(config-if)#ipv6 enable
Ruijie(config-if)#ipv6 address 2001:db8:0:1::cb00:7102/96
Ruijie(config-if)#nat64 enable
Ruijie(config-if)#exit
Ruijie(config)#interface gigabitethernet 0/0
Ruijie(config-if)#ip address 198.51.100.1 255.255.255.0
Ruijie(config-if)#nat64 enable
Ruijie(config-if)#exit
Ruijie(config)#nat64 prefix stateless 2001:db8:0:1::/96
Ruijie(config)#nat64 route 203.0.113.0/24 gigabitethernet 0/1
Ruijie(config)#ipv6 route 2001:db8:0:1::/96 gigabitethernet 0/0 2001:db8:0:1::cb00:7101
Ruijie(config)#ip route 0.0.0.0 0.0.0.0 198.51.100.2
Ruijie(config)#end
```

6) On Host B

Configure the IPv6 address 2001:db8:0:1::cb00:7101/128 on Host B and configure a static route to the prefix 2001:db8:0:1::/96.

7) On Host A

Configure the IP address 198.51.100.2/24 on Host A and configure a static route to the destination network segment 203.0.113.0/24.

Verification

Run the **ping 203.0.113.1** command on Host A.

```
Ping statistics for 203.0.113.1:
   Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
Ruijie#sh nat64 stateless statistics
NAT64 Stateless Global stats:
   Created Packets translation (IPv4 -> IPv6): 0.
   Created Packets translation (IPv6 -> IPv4): 0.
   Packets droped in IPv4: 0.
   Packets droped in IPv6: 0.
NAT64 Stateless Interface stats:
   Gi0/1:
      Created Packets translation (IPv4 -> IPv6): 0.
      Created Packets translation (IPv6 -> IPv4): 0.
   Gi0/0:
      Created Packets translation (IPv4 -> IPv6): 0.
      Created Packets translation (IPv6 -> IPv4): 0.
```

Configuration Example of Multi-Prefix Stateless NAT64

Networking Requirements

Host A with the address 198.51.100.2/24 in the IPv4 domain can access Host B and Host C in different network segments of the IPv6 network. To meet this requirement, configure IPv6 prefixes on the IPv6 interfaces of different network segments on the Stateless NAT64 device (Router B).

Networking Topology

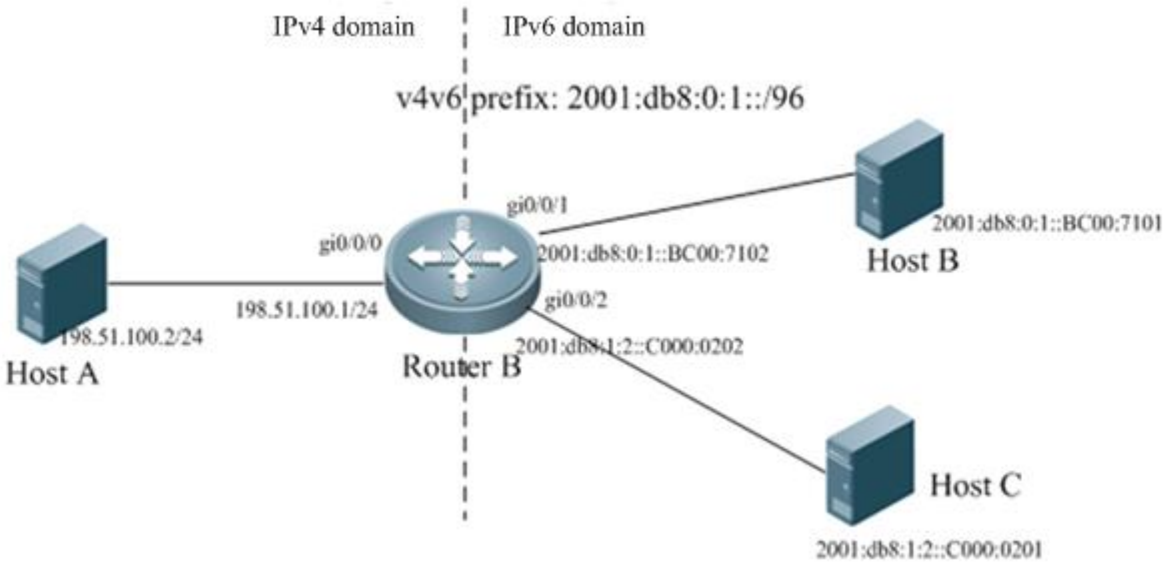


Figure 2-4 Multi-Prefix Stateless NAT64

Host A accesses Host B and Host C in two IPv6 network segments.

Configuration Tips

Perform the following configurations on the Stateless NAT64 device:

- Configure the IPv6 address of an IPv6 network interface, enable NAT64, and configure the Stateless NAT64 prefix (v6v4).
- Configure the IPv4 address of an IPv4 network interface and enable NAT64.
- Configure the Stateless NAT64 prefix (v4v6) in global mode.
- Configure a route which starts from an IPv4 network segment and is destined for the IPv6 interface for NAT64.
- Configure a static route that is used to transmit translated packets to the IPv6 address.

Configuration Steps

8) Perform the following configurations on Router B, which serves as the Stateless NAT64 device.

```
Ruijie#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Ruijie(config)# interface gigabitethernet 0/0/0
Ruijie(config-if)#ip address 198.51.100.1 255.255.255.0
Ruijie(config-if)#nat64 enable
Ruijie(config-if)#exit
Ruijie(config)#interface gigabitethernet 0/0/1
Ruijie(config-if)#ipv6 enable
Ruijie(config-if)#ipv6 address 2001:db8:0:1::cb00:7102/96
Ruijie(config-if)#nat64 enable
Ruijie(config-if)#nat64 prefix stateless v6v4 2011:db8:0:1::/96
Ruijie(config-if)#exit
Ruijie(config)#interface gigabitethernet 0/0/2
Ruijie(config-if)#ipv6 enable
Ruijie(config-if)#ipv6 address 2001:db8:1:2::C000:0202/96
Ruijie(config-if)#nat64 enable
Ruijie(config-if)#nat64 prefix stateless v6v4 2011:db8:1:2::/96
Ruijie(config-if)#exit
Ruijie(config)#nat64 prefix stateless v4v6 2011:db8:2::1/96
Ruijie(config)#nat64 route 203.0.113.0/24 gigabitethernet 0/0/1
Ruijie(config)#ipv6 route 2011:db8:0:1::/96 gigabitethernet 0/0/1 2001:db8:0:1::cb00:7101
Ruijie(config)#nat64 route 0.0.0.0/0 gigabitethernet 0/0/2
Ruijie(config)#ipv6 route 2011:db8:1:2::/96 gigabitethernet 0/0/2 2001:db8:1:2::C000:0201
Ruijie(config)#end
```

9) On Host B

Configure the IPv6 address 2001:db8:0:1::cb00:7101/128 on Host B and configure a static route to the prefix 2001:db8:0:1::/96.

10) On Host C

Configure the IPv6 address 2001:db8:0:1::c000:0201/128 on Host C and configure a static route to the prefix 2001:db8:1:2::/96.

11) On Host A

Configure the IPv6 address 198.51.100.2/24 on Host A and configure a static route to the destination network segment 203.0.113.0/24.

Verification

Run the **ping 203.0.113.1** command on Host A.

```
Ping statistics for 203.0.113.1:
   Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Run the **ping 192.0.2.1** command on Host A.

```
Ping statistics for 192.0.2.1:
   Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
Ruijie#sh nat64 stateless statistics
NAT64 Stateless Global stats:
   Created Packets translation (IPv4 -> IPv6): 16.
   Created Packets translation (IPv6 -> IPv4): 31.
   Packets droped in IPv4: 0.
   Packets droped in IPv6: 0.
```

```
NAT64 Stateless Interface stats:
   Gi0/0/0:
      Created Packets translation (IPv4 -> IPv6): 16.
      Created Packets translation (IPv6 -> IPv4): 0.
   Gi0/0/1:
      Created Packets translation (IPv4 -> IPv6): 0.
      Created Packets translation (IPv6 -> IPv4): 19.
   Gi0/0/2:
      Created Packets translation (IPv4 -> IPv6): 0.
      Created Packets translation (IPv6 -> IPv4): 12.
```