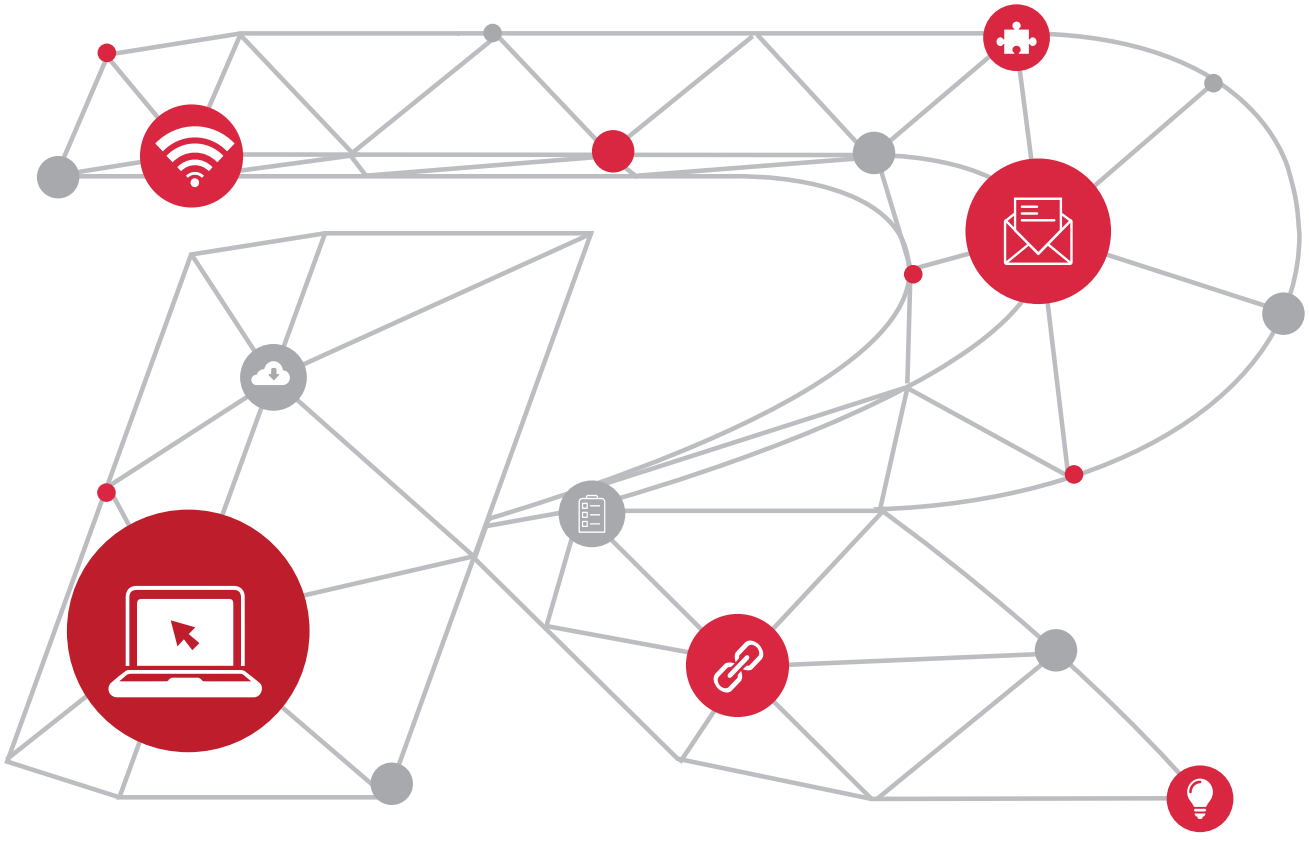
UISS Double-Engine Hot Standby

White Paper



Contents

Introduction

The high availability (HA) requirements cover fault detection and fault recovery. This document focuses on the fault recovery, and describes the technical features, application solutions, and implementation key of the hot standby function based on Ruijie Uninterrupted Supervisor Engine Switchover (UISS) technology.

As the scale and application fields of networks expand rapidly, networks have become an indispensible part of our social life, and enterprises and individuals cannot communicate or share information without networks. As enterprises become more dependent upon networks, the networks must be secure and reliable. Once the network services become unavailable, the productivity and profitability of the enterprises may be significantly affected.

System shutdown falls into two types:

* **Unplanned shutdown (failure)**

An unplanned shutdown is uncontrollable and random, and is generally related to defects in software or hardware.

* **Planned shutdown (maintenance)**

A planned shutdown is used for a purpose such as system maintenance or upgrade. A planned shutdown can be scheduled to minimize the impact on the availability of the system.

Network applications need to transverse multiple network segments. To ensure availability of the network, all the network segments must have a strong fault recovery capability so that the fault recovery is transparent and imperceptible to the users and network applications.

The best way to improve the availability and reliability of a network is to minimize unplanned shutdowns and the MTTR. Even if the system shutdown duration and the MTTR decrease slightly, the availability of the network will be significantly improved.

Therefore, a single point of failure must be avoided so as to ensure a relatively high availability of the network. Globally, redundant links must be available between two communication nodes, and the topology must be managed automatically (for example, layer-2 STP and layer-3 routing protocol) or manually. Locally, network nodes, especially forwarding devices, must be capable of handling a failure of a single component. Redundancy is an important approach to handling a single point of failure.

The HA technology requires that a device have a relatively high reliability. With little room for improvement, the high reliability of components or devices is not the only approach to achieving HA. Generally, the HA objective is achieved by providing redundancy for key components, so that the components can be replaced without shutting down the system. The corresponding software supports hot swapping, fault detection, fault isolation, and fault recovery.

After a fault is detected, if redundant boards adopt cold start, it will take relatively long time to recover (exception handling and cold start). To achieve fast fault recovery, the status of the device must be backed up, which calls for the hot standby function. Real-time status backup ensures that the status of a component is synchronized with that of the corresponding redundant component, thereby ensuring fast fault recovery.

Basics

• Requirements

* **A device that can provide a redundant engine generally adopts the chassis architecture. Distributed functions (that is, the engine is separated from the line card) provides the hardware basis for the HA solution.**

* **A redundant engine is provided to ensure the high availability. When an engine (that is, the active engine) fails or is being maintained, the other engine (that is, the standby engine) takes over the tasks from it.**

* **During the switchover, the physical status and management status of the line card must remain unchanged to ensure nonstop communication.**

From the perspective of communication support, the internal functions of the network device can be divided into data plane, control plane, and management plane. The network device provides the forwarding service for users through the data plane. Therefore, the network is available only if the forwarding function applies properly.

For a single device, the device availability can be enhanced by improving the reliability of the engine. However, it is difficult to meet the HA requirements in this way. Instead, the reliability is ensured by using redundancy design, including:

* **Hardware redundancy**

* **Software support**

A device that can provide a redundant engine generally adopts the chassis architecture. Distributed functions (that is, the engine is separated from the line card) provides hardware basis for the HA solution. The device includes key components such as engine, line card, chassis, fan, and power supply. These components support hot swappable hardware. The engine is responsible for the functions of the management plane and the control plane. The line card provides the functions of the data plane. The chassis connects the engine and the line card. Data forwarding works functionally only if the line card operates properly.

The engine is the management and control center of the device. If the engine fails, the availability of the device will be affected. With the distributed structure, even if the engine fails, the device will not become unavailable immediately. A redundant engine is provided to ensure the high availability. When an engine (that is, the active engine) fails or is being maintained, the other engine (that is, the standby engine) takes over the tasks from it.

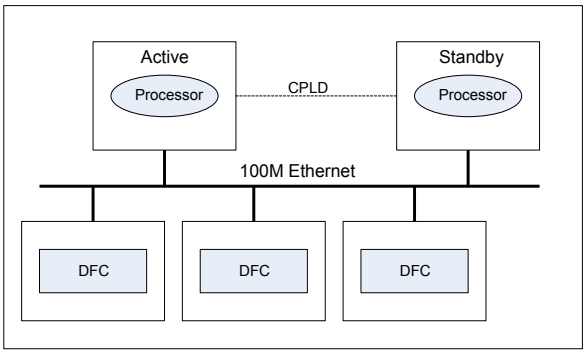
Figure 1 shows the physical model of the hot standby technology based on Ruijie UISS.

During the switchover, the physical status and management status of the line card must remain unchanged to ensure nonstop communication.

When the line card is reloaded, the communication will be interrupted. Even if the line card is not initialized, when the protocol with the topology management function (for example, STP) starts to run, it will generally set the management status of the interface to be disabled. Because the route re-convergence results in route flapping, the communication will be interrupted.

If hot standby is applied, when the redundant board operates in place of the active engine, the line card does not need to be reloaded or reconfigured. Therefore, the data forwarding will not be affected, and the switchover of the supervisor engines is transparent to the communication.

Figure 1 Physical Model of UISS-based Hot Standby Technology



• Keys

The design of the hot standby technology must focus on:

Status Synchronization Between Dual Engines

Information must be collected by the active engine and managed by the standby engine to ensure status synchronization between the active engine and the standby engine.

Status Consistence Within the Standby Engine

Various components in the system are associated with each other in status. However, their status information may be out of synchronization, so the status of the components in the system may be inconsistent. A switchover may occur at any time, and it takes time to detect a fault. In this case, during the system switchover, the status of the components in the system may be inconsistent, and the status recorded by the standby engine may be different from the actual status. Therefore, the status consistency must be checked during the switchover.

Due to complexity of software function, software components vary significantly in construction. Ruijie UISS is designed in a distributed manner, as shown below (technical implementation key):

* **Components (or entities) in the active engine correspond to those in the standby engine in a one-to-one manner, and status synchronization is based on this correspondence, which is called "peer-to-peer synchronization". "Peer-to-peer synchronization" is aimed at ensuring status synchronization between peer entities. It does not ensure status consistence between the components within the standby engine or between the standby engine and the line card.**

* **During the system switchover, nonstop communication is ensured by separating the data plane from the control plane. The foregoing step cannot ensure system status consistence, so during system switchover, the status consistency must be checked, including:**

1. Status consistency check within the standby engine
2. Status synchronization between the standby engine and the line card

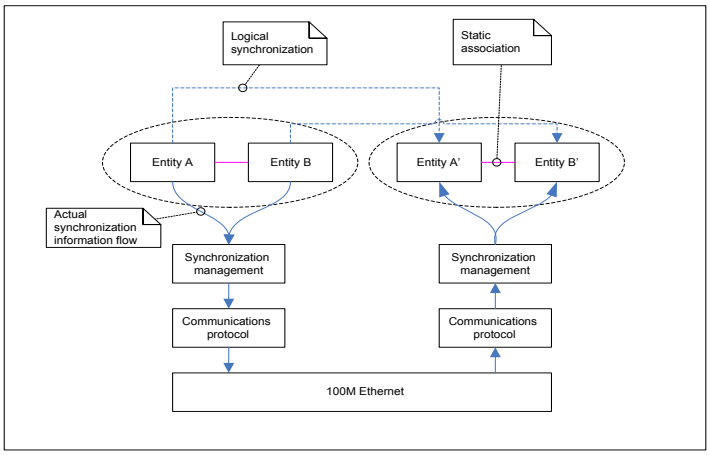
Technical Key 1: Peer-to-Peer Synchronization

The information synchronization of the entities in the active engine targets at the peer entities in the standby engine. The active engine collects information about the entities within the active engine and sends the information to the peer entity. The explanation of the synchronization information makes sense only within entities. The peer entities can realize synchronization in various ways, such as directly sending status information, and sending events received by the entities. The association between the entities is achieved by using a system mechanism at a higher layer.

Peer-to-peer synchronization requires that internal status of entities be always consistent, so an entity must not send its intermediate status to the peer entity. However, it is not necessary to ensure synchronization between the associated entities of the standby engine.

As shown in Figure 2, assuming that entity A is associated with entity B., the status change of entity A will cause the status change of entity B. However, with peer-to-peer synchronization, only the synchronization between A' and A is required, whereas the status consistency between A' and B' is ensured by B. The status inconsistency between A' and B' due to a failure or switchover is handled during system switchover.

Figure 2 Peer-to-peer Synchronization



Technical Key 2: Separating Data Plane from Control Plane (Nonstop Forwarding)

The hot standby technology requires that the communication interruption duration of the data plane be within several milliseconds. However, it takes time for the control plane to recover after the switchover. According to the general operation rules, the old forwarding entries will be deleted during initialization. After the old forwarding entries are deleted, the data forwarding will be interrupted before the new forwarding entries are generated.

After the data plane is separated from the control plane, before a new forwarding table is generated, the old forwarding table is used. The new forwarding table will replace the old forwarding table after the system switchover.

In Ruijie UISS hot standby solution, the dynamic and static ARP entries in the related forwarding table are synchronized, whereas neither the dynamic routing entries nor the dynamic entries in the MAC table are synchronized. However, these entries are cached by using software or hardware in each line card. After the switchover, though a complete routing table is not available in the supervisor engine, the forwarding in each line card continues. In the subsequent convergence process, new entries will be generated by various protocols to replace the old entries in the line card as well as being stored in the routing table on the engine. The old entries that are not replaced after the protocol convergence will be automatically deleted by the line card.

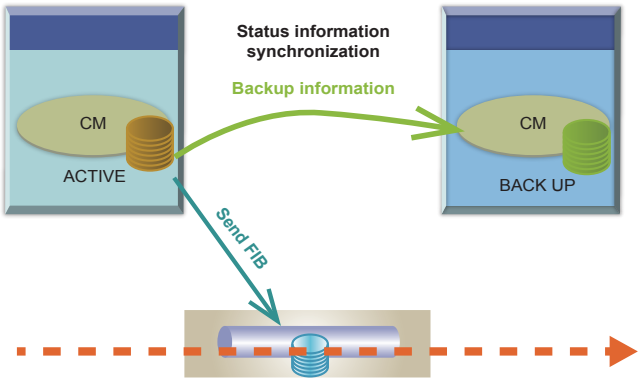
Technical Principle

• Active/Standby Switchover of UISS-based Hot Standby

The hot standby process of the active and standby engines is divided into three phases: **batch status backup**, **real-time status backup**, and **nonstop forwarding**. The batch status backup and real-time status backup are implemented by using the peer-to-peer synchronization technology. Nonstop forwarding is achieved by separating the data plane from the control plane and using the peer-to-peer synchronization technology.

After the secondary active engine starts, the active engine will synchronize all the backup data of the current modules to the standby engine. This process is called batch backup (see Figure 3).

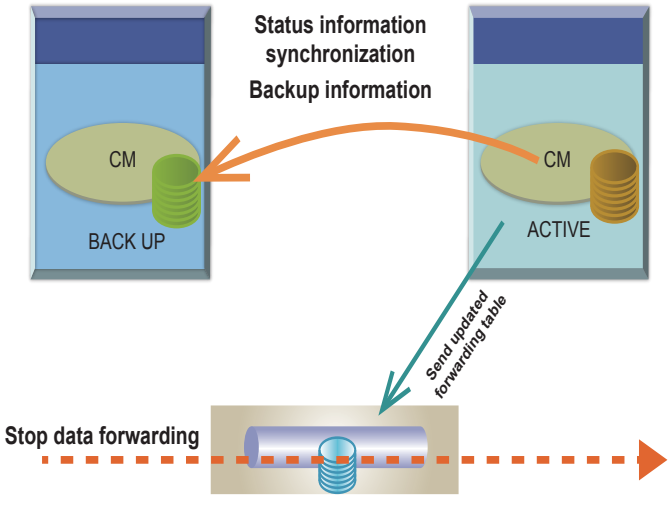
Figure 3 Before the Active/standby Switchover



When the batch backup process is complete, the system enters the real-time backup process. During the real-time backup process, when the backup data of the active engine, the backup data will be synchronized to the standby engine in real time.

After the active/standby switchover, the standby engine becomes an active engine, and informs each module to collect data from the line card and perform a data synchronization. This process is called Nonstop forwarding. During this process, each module actively communicates with the line card to perform verification and synchronization in three aspects: hardware status, link layer status, and configuration data. This ensures consistency in the data and status maintained in the entire system, thereby ensuring proper system operation after the active/standby switchover. When the phase is complete, the new engine becomes a real active engine (see Figure 4).

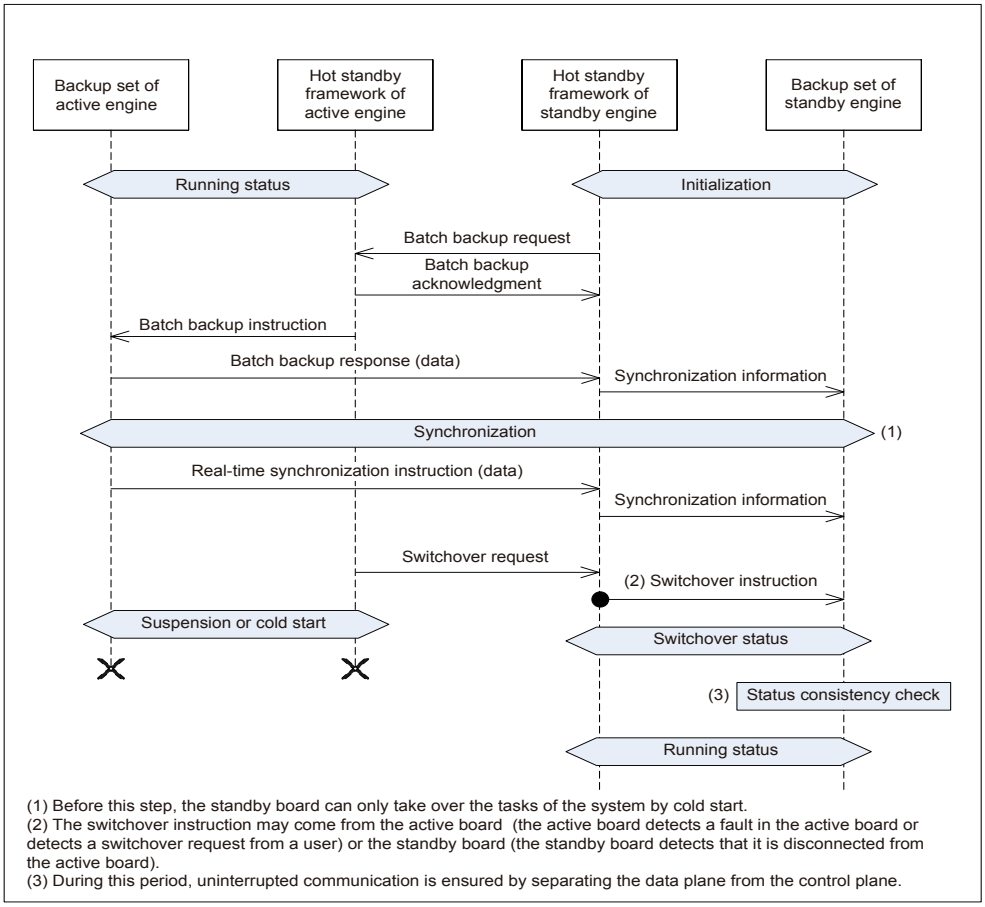
Figure 4 After the Active/standby Switchover



• State Machines of UISS-based Hot Standby

The figure below shows the life cycle model of the UISS hot standby system.

Figure 5 Life Cycle Model of UISS Hot Standby System



The state machines of the active engine transform in this order: waiting for insertion of the standby engine, waiting for the batch backup request, batch backup, real-time backup and nonstop forwarding during switchover.

The state of the standby engine transforms in this order: ready, batch receiving data, and real-time receiving data.

*** After the active engine starts, it is in the state of "waiting for insertion of the standby engine" and waits for the batch request from the standby engine. After the standby engine is initialized, it negotiates the batch backup with the active engine. If a single engine is deployed, the active/standby state machine will stay in this phase.**

*** After the standby engine starts, it is in the state of "ready", and then sends an in-position message to the active engine. After the active engine starts operating properly, the initialized standby engine sends a batch backup request to the active engine.**

*** After the active engine receives the in-position message from the standby engine, the state of the active engine transforms to "waiting for the batch backup request".**

*** After the timer for the state transformation of the standby engine expires, the standby engine actively sends the batch backup request to the active engine, and its state transforms to "batch receiving data". After receiving the batch backup request from the standby engine, the active engine transforms to "batch backup" in state and starts to collect data from each module for synchronization with the standby engine.**

* **When the batch backup of data on each module is complete, the active engine sends message to the standby engine, and its state transforms to "real-time backup". After the standby engine receives the message, its state transforms to "real-time receiving data". After that, the state machines of the active and standby engines are stable.**

1. After receiving the synchronization information, the standby engine is only responsible for the configuration and status synchronization for the entities in the standby engine, and does not send its status change to associated entities as an event.

2. The standby engine adopts the peer-to-peer synchronization approach. Each entity in the backup set of the active engine only backs up its own configuration or status to the peer, and is not responsible for status synchronization for associated entities.

* **When the standby engine detects that the active engine is not operating normally, the state of the active engine transforms to "nonstop forwarding". During the smooth switchover process, the new active engine will collect data from the line card for synchronization. After the process is complete, it becomes the active engine, and its state machine transforms to "waiting for insertion of the standby engine". After the original active engine restarts, it becomes the standby engine, and its initial state is "ready".**

• Active/Standby Switchover from Standby Engine to Active Engine

The active/standby switchover may occur in the following cases:

* **The active/standby switchover command is executed to force a switchover.**

* **The active engine performs a cold reset or is manually removed, which results in the active/standby switchover.**

* **The active engine restarts because of software failure, which results in the active/standby switchover. For example, the hardware watchdog restarts as a module occupies CPU for a long period, or the system restarts due to exceptional data/ instruction access.**

The switchover between the supervisor engines includes the following key processes:

1. Status of the software entities is constant

The user only needs to run the configuration again to synchronize the configuration.

For the status synchronization, the focus should be on ensuring status consistency between the associated entities.

* **The upper-layer entity re-triggers the status-related event, so that the status of the lower-layer entity is consistent with that of the upper-layer entity after the lower-layer entity processes the event. If the lower-layer entity triggers an event for multiple times and generates different results, the status must be cleared before the switchover.**

* **An entity actively checks the status of the entity on which it relies.**

2. Status synchronization between software and hardware

The following approaches are available:

* **The hardware-related configuration of the standby engine is reconfigured on the line card.**

* **The line card sends its buffered status to the engine.**

* **If some status in the line card is not synchronized by using software, after the switchover, the engine performs a recalculation based on the current status, and set the new status in the DFC to replace the old status. The old status not replaced will be automatically discarded after being kept by the DFC for a certain period. In this way, the data plane is separated from the control plane to ensure nonstop communication.**

The foregoing methods require the line card to adopt a distributed solution and buffer the configuration and status related to the data forwarding of the line card.

• Registration Mechanism

During switchover of the state machine, each software module should perform corresponding processing. Each software module uses the registration mechanism to report the processing to the system. In case of status switchover, the called modules are processed according to their priorities.

List of Software Modules

No.	Module	Description
1	Anti IP address scan	Scan attack prevention
2	TPP	Topology protection protocol
3	VLAN	GVRP VLAN learning and configuration management modules
4	AP	Port aggregation module
5	SPAN	Port mirroring
6	IP+MAC	IP+MAC address binding
7	DoS Protection	DoS attack prevention module
8	Port security	Security port module
9	Storm control	Unicast, multicast, and broadcast data control
10	CPP	CPU protection technology
11	ACL/QoS	ACL and QoS modules
12	Routing	Policy-based routing and static routing (IPv4 and IPv6) module
12	STP/RSTP/MSTP	Spanning tree modules
14	IGMP Snooping	IGMP snooping module
15	ARP	ARP protocol and entries
16	Static address table, filtering address table	Address filtering module

Performance

• Configuration Layer

The configuration of the active engine is backed up to the standby engine through the batch backup and real-time backup processes. When the preparation is ready, all configuration of the active engine has been stored on the standby engine and synchronized. Therefore, during the active/standby switchover, the configuration layer can achieve smooth transition without collecting or synchronizing other data.

• Protocol Layer

To ensure nonstop forwarding of service data, it is not plausible to re-learn the various forwarding tables after deleting them. During the data collecting and synchronization processes performed by the active engine, the only changed data of the line card is updated and refreshed while the other data remain unchanged.

Layer-2 Unicast

In general layer-2 unicast forwarding, only the MAC address table is used, and the MAC address table related to the packet forwarding is available on the service board. When the active/standby switchover occurs, the new active engine will initiate collecting and synchronization of MAC address to the service board, but the original MAC address table on the service board will not be deleted. This ensures nonstop forwarding of layer-2 unicast data.

Layer-2 Multicast

Similarly, for the layer-2 multicast, the multicast MAC entries required by forwarding are stored on the service board. When the active/standby switchover occurs, the active engine collects data from the interface board, whereas the original multicast MAC entries on the service board remain unchanged. This ensures nonstop forwarding of layer-2 multicast flows.

Layer-3 Unicast

For the layer-3 unicast, the forwarding primarily relays on ARP and FIB of the interface board. During the smooth active/standby switchover, the original ARP and FIB on the service board remain unchanged. This ensures nonstop forwarding of layer-3 unicast flows.

Layer-3 Multicast

Similarly, for the layer-3 multicast, during the smooth switchover, the original multicast entries on the interface board remain unchanged. This ensures nonstop forwarding of the original multicast flows.

Conclusion

Ruijie UISS hot standby solution ensures the switchover between the active and standby engines can be implemented within 3 seconds. In addition, during the switchover, each line card in the active engine can forward data flows without affecting transparent operation of the network services. This ensures smooth switchover between the supervisor engines, thereby significantly improving device reliability and network availability.