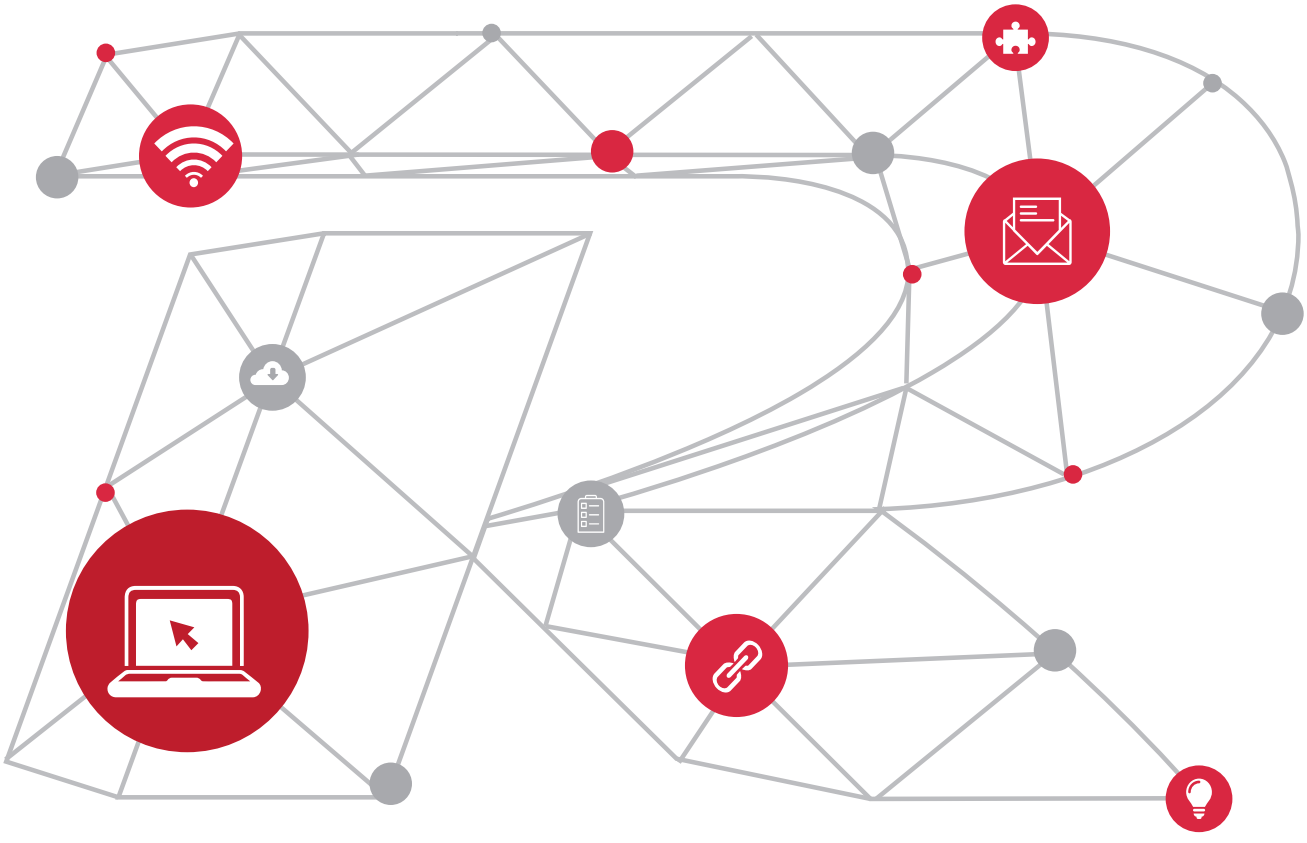
Ruijie CPU Protection Policy

White Paper



Contents

Introduction

The CPU Protection Policy (CPP) is a method for enhancing switch security. By differently processing packets destined for the CPU, the CPP protects the processor and channel bandwidth resources of the switch, and ensures normal packet transfer and normal protocol statuses.

• Packets Destined for the CPU

Most flows received by a switch undergo hardware processing via switching chips and do not require the CPU resources. Some other flows must be specially processed by the CPU or require the participation of the CPU. These flows contain various special packets, such as management packets and protocol packets, which must be processed by the CPU for device maintenance and normal network operation. Besides, IP packets whose routes are not established require the CPU to participate in the processing to obtain information about the next hop.

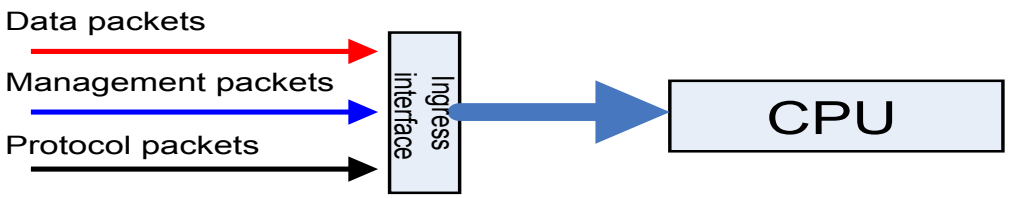
Packets destined for the CPU can be classified into the following types by packets usage and importance.

1. **Management Packets** are used for the detection and maintenance of multiple stacked devices or of devices between the supervisor module and line card, and used for the management and configuration of remote devices. Such packets include stacking management packets, SNMP packets, and Telnet configuration and management packets. These packets are necessary and common.

2. **Protocol Packets** are used to maintain protocols, including BPDU packets, GVRP packets, OSPF packets, and so on. These packets are critical for network maintenance.

3. **Data Packets** include IP packets without routes established, IP option packets, packets whose TTL is 0 or 1, and L3 IP header checksum error packets.

Figure 1 Packets Destined for the CPU

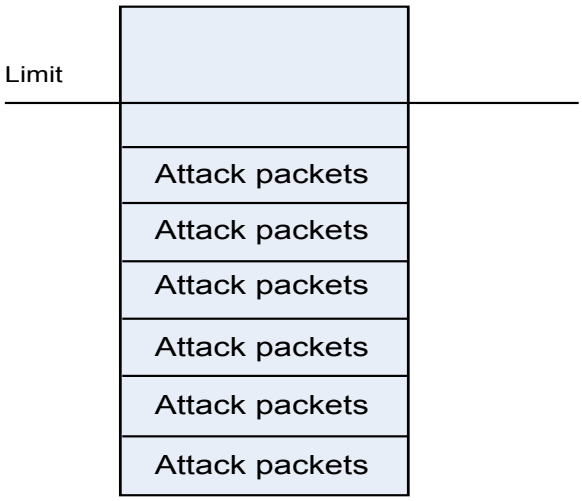


• Existing Security Defects

The CPU does not distinguish network flows in these three types and processes them in the same way. The CPU processes flows by using a certain scheduling algorithm (such as FIFO). Generally, the CPU can process flows in time. However, if there are too many flows, or attackers send a large number of attack packets, the CPU will be busy processing the packets, which leads to high CPU utilization, or even discarding of other packets. This affects switch use and network stability. In this application environment, the switch experiences great security risks, leaving opportunities for attackers.

In one attack method, attackers can send a large number of ARP requests to the switch, and as a result, it is possible that telneting to the switch fails, because the switch is busy processing the ARP request packets and cannot process Telnet packets in time. Another typical attack is ping flooding attack. This attack can also cause high CPU utilization and affect normal operation of other functions. Figure 2 shows the CPU resource utilization upon an attack.

Figure 2 CPU Resource Utilization



As shown in Figure 2, **Limit** indicates the upper limit of CPU resources. At this time, CPU resources are all occupied by attack packets. Other packets will be discarded because of no resources. Attacks against the CPU mainly cause the following impact on the switch:

1. The CPU utilization is excessively high, and can reach even 100%.

2. Non-attack packets are not processed in time, causing network disconnection and topology flapping and damaging stability of the whole network.

3. The CLI function is slow in response, or even fails to respond.

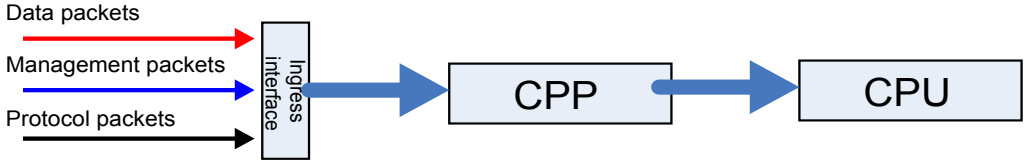
In this processing method, the switch CPU directly connects to the outer network without necessary protection, leading to a poor in attack-defense capability.

Working Principle

For such defects, Ruijie has developed the CPP technology to effectively protect the switch CPU, limit and even prevent attack packets, ensure that required packets are normally processed with priority upon attacks, and guarantee the stable operation of the switch and network.

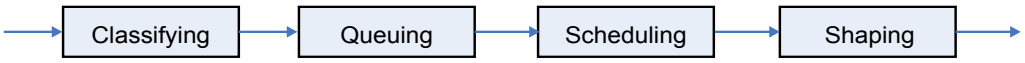
The basic principle of the CPP is that the technology distinguishes between flows destined for the CPU, queues the flows based on priorities, and implements bandwidth limit to fully ensure that the CPU will not be occupied by invalid traffic or maliciously attacked, and resources will not be used up. Figure 3 shows the CPP process.

Figure 3 CPP Process



The CPP identifies and classifies all packets destined for the CPU in accordance with a certain mechanism, and then differently processes the packets of each type. Modules inside the CPP can be divided into the processes shown in Figure 4.

Figure 4 Internal Processes of the CPP



Classifying: Flows must be classified before they can be differently processed. In this process, a packet is identified in accordance with a certain mechanism and then classified into a defined type according to the classification rules.

Queuing: Packets in different types are mapped to different queues according to the configuration. Each queue has a different transmission priority. By classifying and queuing, all flows are distinguished.

Scheduling: When flows in multiple queues are waiting for transmission, a flow in a queue is selected according to the scheduling algorithm. In the CPP, scheduling is strictly implemented by priority, so that queues with higher priorities can be preferentially transmitted. To be specific, queue 7 has the highest priority, followed by queue 6 and so on. Packets in a queue with a higher priority are preferentially transmitted.

Shaping: This process is used for the maximum bandwidth control of one flow (or more flows) on the egress and/or the maximum bandwidth control of the whole port. In each priority-based queue, the rate of packets entering the queue is controlled according to the configured maximum bandwidth. At a certain moment, packets with a rate exceeding the rate limit will be discarded. The switch can implement rate limit in the following modes:

1. Flow-based rate limiting: The rate is limited by the packet type. Packets with a rate exceeding the rate limit will be discarded.

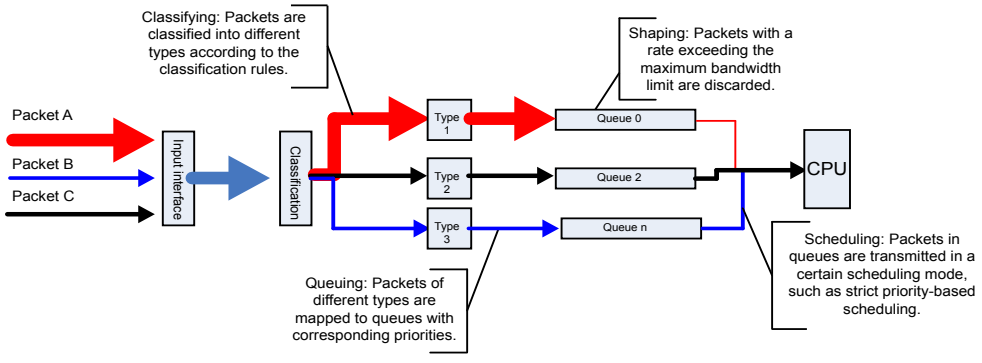
2. Queue-based rate limiting: The rate is limited for packets to be received in each queue. Packets received in the queue with a rate exceeding the rate limit will be discarded.

3. CPU-based rate limiting: The overall receiving rate of the CPU is limited. Packets with a receiving rate exceeding the rate limit will be discarded.

When one or more of the three rate limit modes are applied on the switch, only packets meeting all applied rate limit requirements will not be discarded.

By using this distinguishing mechanism, packets of different Apps are mapped to different queues, and important packets are preferentially processed according to the queue scheduling algorithm. Meanwhile, bandwidth limits are applied to the queues or packets to prevent packets from occupying too many resources. In CPP mode, packets destined for the CPU follow the process shown in Figure 5.

Figure 5 Process in CPP Mode



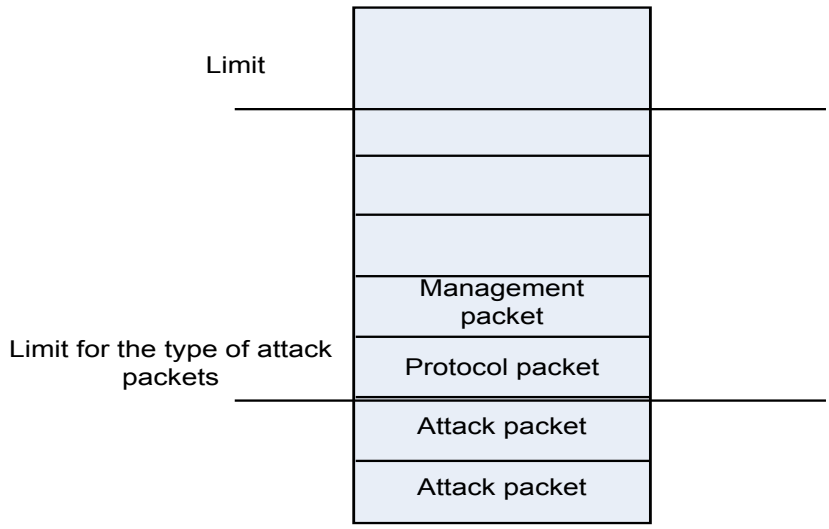
As shown in Figure 5, assume that packet A is an attack packet, packet B is a management packet, and packet C is a protocol packet. The attack packet can be prevented in the following ways:

1. Map the type of the attack packet to a queue with a low priority, such as queue 0. Simultaneously, map the type of the management packet to a queue with the highest priority, and that of the protocol packet to a queue with the second highest priority.

2. Set the maximum bandwidth value for each queue: Set the maximum bandwidth of the queue of the attack packet to a small value, and those of other queues to greater values.

After the above settings, packets in queues with higher priorities are preferentially transmitted because strict priority-based scheduling is adopted. As a result, the management packet is processed first, followed by the protocol packet, and finally the attack packet. In addition, if traffic of attack packets in queue 0 exceeds the configured bandwidth limit, and excess packets are discarded. In this mode, although there are a large number of attack packets, management packets and protocol packets can still be preferentially processed, which guarantees the stable operation of the switch and network. Moreover, most attack packets are discarded, ensuring that switch resources are not occupied by attack packets in a large scale. In CPP mode, the CPU resource utilization is shown in Figure 6.

Figure 6 CPU Resource Utilization in CPP Mode



As shown in Figure 6, **Limit** indicates the upper limit of CPU resources. A CPU resource use limit is also configured for the type of attack packets. In CPP mode, CPU resources occupied by attack packets are limited. This ensures the preferential processing and CPU resource requests of other packets.

The CPP can be implemented by hardware or by combining hardware and software. The software mainly implements secondary classification of packets to sort out packets to be separately controlled, and limits the bandwidth for each type of packets to determine packet discarding in advance and reduce packet processing time on the CPU. Such implementation can enhance the CPU packet processing performance and reduce the CPU work load as much as possible.

Technical Features

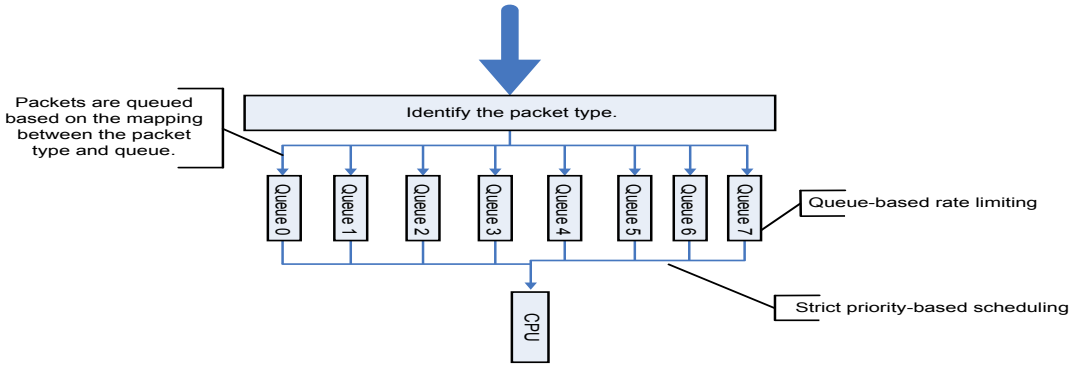
The CPP can be implemented by only hardware or by a combination of hardware and software.

• Implementation Based on Only Hardware

1. Type-based management: One or more types of packets destined for the CPU are classified into a same type according to the identification mechanism inside the hardware. In the queue mapping relationship, each type can map to any of the eight queues. An operation performed on a certain type affects all packets of this type. For example, if a certain type is mapped to a queue, all packets of this type are mapped to this queue.

2. Queue-based rate limiting: This type of rate limiting is performed in the unit of kbps, and cannot separately limit the rate of a certain packet type.

Figure 7 CPP Implementation Based on Only Hardware



In the process of sending packets to the CPU, the packets will be mapped to the corresponding queues based on the user configuration result. When the rate of a queue exceeds the maximum limit, excess packets will be discarded.

In the case of an attack, the type of the attack packet is determined and mapped to a queue with a low priority, such as queue 0. The bandwidth for the queue is set to a small value so that impact of the attack packet on the switch can be effectively limited.

Advantage: Flow classification and rate limiting are implemented by hardware.

Disadvantage: The flow classification and rate limiting granularity are limited by the hardware capability.

• Implementation Based on a Combination of Software and Hardware

1. Packets are classified in a relatively refined manner via ACLs based on L2, L3, and L4 packet information, and packet-based mapping is available. Most packets can be classified by hardware. If some packets cannot be classified by hardware due to hardware limits, secondary classification can be implemented by software. In this way, all types of packets can be distinguished as required. When a packet matches the ACL, the packet priority can be changed so that the mapping between the packet and a queue is changed.

2. Packet-based rate limiting: The rate of each type of packets can be separately limited on the hardware and software in the unit of PPS.

3. Rate limiting and filtering can be performed multiple times to ensure switch security. The CPP function is available on both the smart card and supervisor module, which not only reduces the processing load of the supervisor module, but also ensures the correctness of the CPP function.

Advantage: The flow classification capability is stronger.

Disadvantage: More CPU resources are consumed.

Figure 8 shows the process of CPP implementation based on a combination of software and hardware.

Figure 8 Process of CPP Implementation Based on a Combination of Software and Hardware

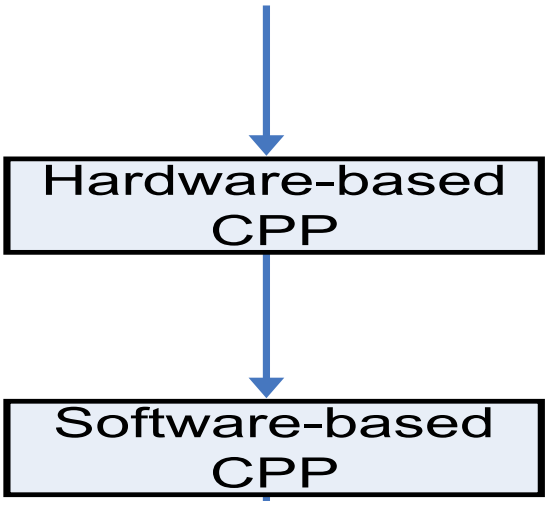


Figure 9 Hardware-based Implementation

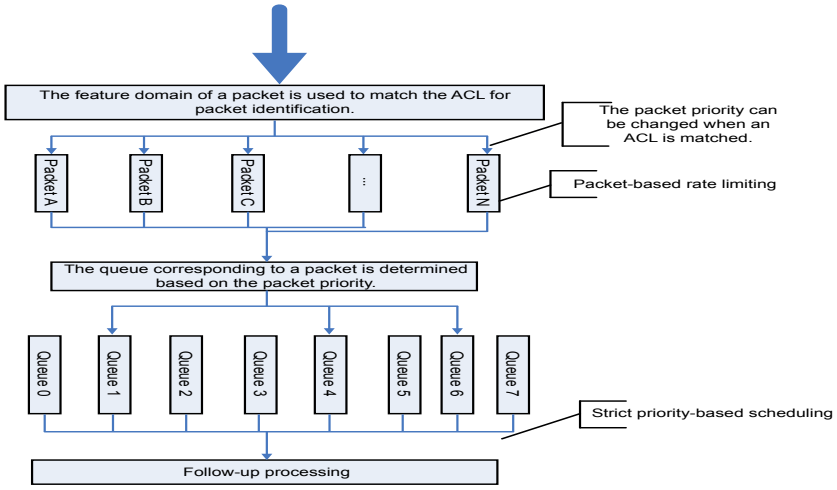
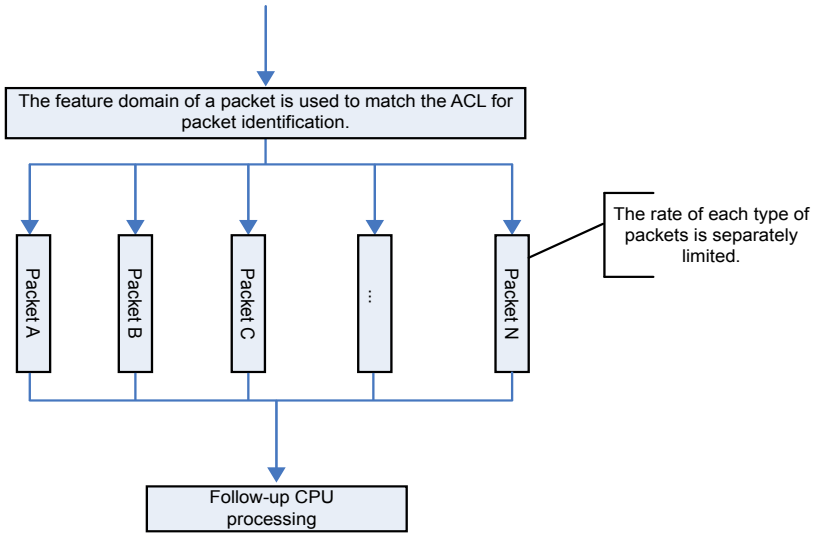


Figure 10 Software-based Implementation



In the case of an attack, the priority of the attack packet is set to a small value, and the rate of the attack packet is limited, so that the attack packet is limited on the hardware or software, thereby effectively limiting impact of the attack packet on the switch.

Analysis of Practical Application

S86 series switches are used as an example to test the CPP function. Tables below list impact of different types of attacks on the switch with the CPP enabled and disabled.

1. ARP attack packets

Quantity of Source IP Addresses	Sending Rate	Ping Latency	VRRP Flapping or Not	STP Flapping or Not	OSPF Flapping or Not	PIM Flapping or Not
CPP Disabled (Inside the Card)						
1	148 PPS	< 127 ms, with 1 or 2 packets lost occasionally	Disabled	Disabled	No	No
	1488 PPS	< 113 ms, with 4 packets lost	Disabled	Disabled	No	No
	14881 PPS	< 131 ms, with 4 packets lost out of 20	Disabled	Disabled	No	No
	148810 PSS	< 112 ms 4,382 ms required from a ping failure to a ping success	Disabled	Disabled	No	Yes
10	148 PSS	< 127 ms, with 1 or 2 packets lost occasionally	Disabled	Disabled	No	No
	1488 PSS	< 113 ms, with 4 packets lost	Disabled	Disabled	No	No
	14881 PSS	< 131 ms, with 4 packets lost out of 20	Disabled	Disabled	No	No
	148810 PSS	< 112 ms < 4,382 ms required from a ping failure to a ping success	Disabled	Disabled	No	Yes
CPP Limiting ARP Rate Within 500 PPS (Inside the Card)						

Quantity of Source IP Addresses	Sending Rate	Ping Latency	VRRP Flapping or Not	STP Flapping or Not	OSPF Flapping or Not	PIM Flapping or Not
1	148 PSS	< 89 ms	Disabled	Disabled	No	No
	1488 PSS	< 100 ms	Disabled	Disabled	No	No
	14881 PSS	< 121 ms	Disabled	Disabled	No	No
	148810 PSS	< 133 ms	Disabled	Disabled	No	No
10	148 PSS	< 127 ms, with 1 or 2 packets lost occasionally	Disabled	Disabled	No	No
	1488 PSS	< 113 ms, with 4 packets lost	Disabled	Disabled	No	No
	14881 PSS	< 131 ms, with 4 packets lost out of 20	Disabled	Disabled	No	No
	148810 PSS	< 112 ms < 4,382 ms required from a ping failure to a ping success	Disabled	Disabled	No	Yes
CPP Limiting ARP Rate Within 500 PPS (Inside the Card)						
1	148 PSS	< 89 ms	Disabled	Disabled	No	No
	1488 PSS	< 100 ms	Disabled	Disabled	No	No
	14881 PSS	< 121 ms	Disabled	Disabled	No	No
	148810 PSS	< 133 ms	Disabled	Disabled	No	No
10	148 PSS	< 89 ms	Disabled	Disabled	No	No
	1488 PSS	< 100 ms	Disabled	Disabled	No	No
	14881 PSS	< 121 ms	Disabled	Disabled	No	No
	148810 PSS	< 133 ms	Disabled	Disabled	No	No

The table above indicates that when the CPP function is disabled, some ping packets are lost, and with a relatively high ARP attack rate, PIM flapping occurs; when the CPP function is enabled, and the rate of ARP packets is limited, neither ping packet loss nor flapping occurs. The CPP function properly ensures the switch resources and stability in the case of ARP attacks.

2. Layer-2 protocol packet attacks (BPDU attacks)

Protocol Enabled/ Disabled	Sending Rate	Ping Latency	VRRP Flapping or Not	STP Flapping or Not	OSPF Flapping or Not	PIM Flapping or Not
CPP Disabled (Inside the Card)						
MSTP enabled	148 PSS	< 100 ms, with a packet loss rate of 10%	Disabled	No	Yes	No
	1488 PSS	< 216 ms, with a packet loss rate of 40%	Disabled	No	Yes	Yes
	14881 PSS	< 580 ms, with a packet loss rate of 75%	Disabled	Yes	Yes	Yes
	148810 PSS	< 80 ms, with a packet loss rate of 90%	Disabled	Yes	Yes	Yes
MSTP disabled	148 PSS	< 184 ms, with 2 packets lost out of 10	Disabled	Disabled	No	No
	1488 PSS	< 127 ms, with a packet loss rate of 15%	Disabled	Disabled	No	No
	14881 PSS	< 147 ms, with a packet loss rate of 25%	Disabled	Disabled	No	No
	148810 PSS	< 128 ms, with a packet loss rate of 40%	Disabled	Disabled	Yes	No
CPP Limiting BPDU Rate Within 100 PPS (Inside the Card)						
MSTP enabled	148 PSS	< 137 ms, with 1 or 2 packets lost occasionally	Disabled	Disabled	No	No
	1488 PSS	< 117 ms, with a packet loss rate of 3%	Disabled	Disabled	No	No
	14881 PSS	< 100 ms, with a packet loss rate of 5%	Disabled	Disabled	No	No
	148810 PSS	< 142 ms, with a packet loss rate of 5%	Disabled	Disabled	No	No

Protocol Enabled/ Disabled	Sending Rate	Ping Latency	VRRP Flapping or Not	STP Flapping or Not	OSPF Flapping or Not	PIM Flapping or Not
MSTP disabled	148 PSS	< 128 ms, with 1 or 2 packets lost occasionally	Disabled	Disabled	No	No
	1488 PSS	< 144 ms, with 1 or 2 packets lost occasionally	Disabled	Disabled	No	No
	14881 PSS	< 104 ms, with 3 or 4 packets lost occasionally	Disabled	Disabled	No	No
	148810 PSS	< 116 ms, with 3 or 4 packets lost occasionally	Disabled	Disabled	No	No

In the case of BPDU attacks, if the CPP function is disabled, the switch is greatly affected, a considerable number of ping packets are lost, and STP, OSPF, and PIM flapping occur. After the CPP function is enabled, and the rate of BPDU packets is limited, few ping packets are lost, and no protocol flapping occurs. The CPP function properly ensures the switch resources and stability in the case of BPDU attacks.

3. Layer-3 protocol packet attacks (RIP attacks)

Protocol Enabled/ Disabled	Sending Rate	Ping Latency	VRRP Flapping or Not	STP Flapping or Not	OSPF Flapping or Not	PIM Flapping or Not
CPP Disabled						
RIP enabled	148 PSS	< 98 ms, with 1 packet lost occasionally	Disabled	Disabled	No	No
	1488 PSS	< 98 ms, with 2 or 3 packets lost occasionally	Disabled	Disabled	No	Yes
	14881 PSS	< 122 ms, with a packet loss rate of 25–40%	Disabled	Disabled	No	Yes
	148810 PSS	< 125 ms, with a packet loss rate of 40–50%	Disabled	Disabled	No	Yes

Protocol Enabled/ Disabled	Sending Rate	Ping Latency	VRRP Flapping or Not	STP Flapping or Not	OSPF Flapping or Not	PIM Flapping or Not
CPP Limiting RIP Packet Rate Within 500 PPS						
RIP enabled	148 PSS	< 80 ms	Disabled	Disabled	No	No
	1488 PSS	< 109 ms	Disabled	Disabled	No	No
	14881 PSS	< 148 ms, with 3 packets lost	Disabled	Disabled	No	No
	148810 PSS	< 78 ms, with 5 packets lost	Disabled	Disabled	No	No

In the case of RIP attacks, if the CPP function is disabled, a relatively large number of ping packets are lost, and PIM flapping occurs. After the CPP function is enabled, and the RIP rate is limited, few ping packets are lost, and no flapping occurs. The CPP function properly ensures the switch resources and stability in the case of RIP attacks.

4. Attacks with packet TTL = 1

Protocol Enabled/ Disabled	Sending Rate	Ping Latency	VRRP Flapping or Not	STP Flapping or Not	OSPF Flapping or Not	PIM Flapping or Not
CPP Disabled						
Unicast packets with TTL = 1	148 PSS	< 112 ms	Disabled	Disabled	No	No
	1488 PSS	< 121 ms, with 1 or 2 packets lost occasionally	Disabled	Disabled	No	No
	14881 PSS	< 106 ms, with 2 or 3 packets lost occasionally	Disabled	Disabled	No	Yes
	148810 PSS	< 56 ms, with a packet loss rate of 50%	Disabled	Disabled	Yes	Yes

Protocol Enabled/ Disabled	Sending Rate	Ping Latency	VRRP Flapping or Not	STP Flapping or Not	OSPF Flapping or Not	PIM Flapping or Not
CPP Limiting Rate of Packets with TTL = 1 Within 500 PPS						
Unicast packets with TTL = 1	148 PSS	< 102 ms	Disabled	Disabled	No	No
	1488 PSS	< 107 ms	Disabled	Disabled	No	No
	14881 PSS	< 113 ms, with 1 or 2 packets lost	Disabled	Disabled	No	No
	148810 PSS	< 127 ms, with 1 or 2 packets lost occasionally	Disabled	Disabled	No	No

In the case of an attack with packet TTL = 1, if the CPP function is disabled, a relatively large number of ping packets are lost, and PIM and OSPF flapping occur. After the CPP function is enabled, few ping packets are lost, and no flapping occurs. The CPP function properly ensures the switch resources and stability in the case of attacks with packet TTL = 1.

The above data comparison shows that, in cases of various attacks, the CPP function can properly prevent switch resources from being occupied by attack packets, and can differently process the packets, thereby protecting basic protocols from flapping and status transition, effectively enhancing the switch stability, and greatly improving the capability of processing abnormal packets on the network.

CPP Configuration Principles

CPP-related configuration includes the configuration of the mapping between a packet type and a queue (or a packet priority), bandwidth for the queue (or packets), and bandwidth limit of the CPU. A packet classification mechanism based on packet importance is provided, which classifies common and important packet types such as BPDU and ARP packets into an independent type for separate control, and classifies other packets. As switches support different functions, see corresponding configuration guides of the switches for details about classification and configuration commands.

• Priority Configuration Principles

A corresponding priority can be configured for each type of packets, and the hardware can automatically send the packets of this type to the queue with the specified priority. Priority-based queues are scheduled strictly according to the priority algorithm. To be specific, priority 7 is the highest priority, and packets in the queue with this priority will be transmitted first. Priority 6 is the second highest priority, and so on. Packets in a queue with a higher priority are preferentially transmitted than packets in a queue with a lower priority. Although the mapping between a packet type and a queue is not limited, abnormal mapping configuration may affect normal operation of the switch. For example, if management packets are mapped to queue 0, and the bandwidth is very small, the management packets may not be processed in time, or even may not be processed, which affects device stability. The following criteria for defining packet priorities are recommended for accurate mapping configuration:

1. The priority of management packets is the highest, and the priority of protocol packets is higher than that of data packets.

* **Management packets such as Telnet configuration management packets and SNMP packets are used for management and configuration of remote devices.**

* **Data packets include forwarded packets and invalid packets.**

* **The destination of protocol packets can be layer 2 and layer 3 of the local device.**

Each packet type can be further divided into several sub-types, and priorities are defined for these sub-types according specific criteria.

2. In protocol packets, the priority of layer-2 packets is higher than that of layer-3 packets.

* **Layer-2 protocol packets include BPDU packets, GVRP packets, and protocol packets reserved for internal use.**

* **Layer-3 protocol packets include OSFP, VRRP, PIM, RIP, IGMP packets, and so on.**

* **The priority of ARP packets is to the same as that of layer-3 protocol packets.**

3. The priorities of protocol packets depend on the dependency between protocols.

* **For example, for layer-3 protocol packets with dependency between protocols, the priority of packets of a protocol which is depended on is higher than that of a dependent protocol.**

* **Basic protocol packets such as IGP, BGP, OSPF, and RIP packets are routed at layer 3.**

* **Application protocol packets such as VRRP, IGMP, PIM, and DVMRP packets are routed at layer 3. Among them, VRRP packets have a higher priority because other packets depend on VRRP stability.**

4. The priority of valid data packets is higher than that of invalid data packets.

* **Invalid packets include TTL error packets, option error packets, and L3CRC error packets.**

* **Valid packets include TCP/UDP data packets, IP option packets, and so on.**

Here, only common criteria are provided for priority mapping, and actual priority mapping criteria can be determined based on the network environment. For example, packets of several types with different priorities can be mapped to one queue based on the actual switch resource status and resource overheads. By default, a switch uses the recommended configuration to map various packets.

Conclusion

Ruijie CPP function allows packets destined for the CPU be flexibly differentiated and processed, ensuring normal operation of various services.

Ruijie Networks Co.,Ltd

For further information,please visit our website **http://www.ruijienetworks.com**