Contents

1 Configuring CPP

1.1 Introduction

Many malicious attacks exist in the network environment. CPU attack packets significantly increase the CPU usage of a device, which affects the CPU to process and respond to the CPU to normal services and thus causes device function exceptions. CPU protect policy (CPP) protects the CPU of a device. By controlling the traffic of different types of packets sent to the CPU of the device and scheduling their priorities, users can defend against malicious attacks in the network and ensure that the CPU processes normal protocol packets.

1.1.1 Basic Concepts

1. **QoS**

   As a network security mechanism, Quality of Service (QoS) is a technology that is used to solve network delay and congestion.

   Differentiated Service (DiffServ) is a typical model of QoS that provides different services to packets by classifying service streams.

2. **Bandwidth**

   Bandwidth specifies the maximum data transmission rate and the rate threshold for packet transmission in this document. When the data transmission rate exceeds the bandwidth, packets beyond the bandwidth are discarded. Therefore, the transmission rates of different types of packets must be equal to or smaller than the bandwidth. Both the bandwidth and data transmission rate use the unit of packets per second (pps).

3. **L2/L3/L4**

   L2/L3/L4 means a hierarchical structure of packets based on the TCP/IP model. L2 means layer-2 headers, namely, the Ethernet encapsulation part; L3 means layer-3 headers, namely, the IP encapsulation part; L4 means layer-4 headers, usually, the TCP/UDP encapsulation part.

4. **Priority Queue**

   The strict priority (SP) queue scheduling algorithm is a QoS scheduling algorithm. This scheduling algorithm requires packets to be processed based on the queue priority from high to low. The CPU must process first the packets in a high-priority queue and then those in a low-priority queue.

   Packets to be sent to the CPU are cached in the SP queue and scheduled according to the algorithms of the SP queue. Packet scheduling is the process of selecting and transmitting packets in a priority queue to the CPU.

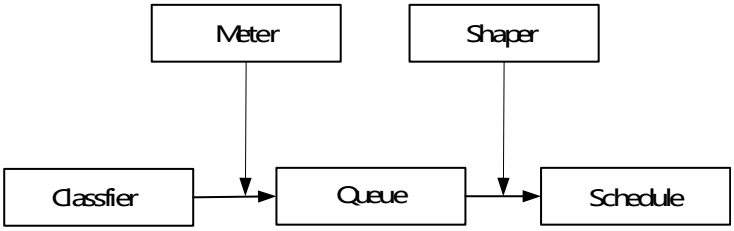
5. **CPU Interface**

   A device caches packets before it sends them to the CPU. Sending the packets to the CPU is similar to a packet output process from the port. The CPU port is a virtual port through which packets are sent to the CPU. The priority queue and SP queue scheduling algorithms apply to only data packets that are transmitted through this CPU port.

1.1.2 Processing Process

The CPP function protects the CPU by using the standard QoS DiffServ model.

Figure 1-1 CPP Implementation Model



1. Classifier

The classifier classifies all packets to be sent to the CPU by the L2, L3 and L4 information of the packets. Classifying packets is the basis for implementing QoS policies. In subsequent actions, CPP can, according to the classification, implement different policies and provide different services. Packets on a device are classified by the supported protocols and management functions, for example, Bridge Protocol Data Unit (BPDU) packets of Spanning Tree Protocol (STP) and packets of Internet Control Message Protocol (ICMP). Packet types cannot be customized.

Instruction

Packet classification varies with products due to hardware difference. For details, see *Specifications* for details.

2. Meter

The meter limits the rates of different types of packets based on the set rate thresholds. You can set different rate thresholds for various packet types. When the rate of a packet type exceeds the threshold, the packets beyond the threshold are discarded.

By using the meter, you can control the rate of a packet type sent to the to the CPU within a threshold to prevent specific attack packets from exerting great impacts on the CPU resources. This is the level-1 protection of CPP.

3. Queue

Queues are used to classify packets at level 2. You can select the same queue for different packet types; queues cache packets in a device and provide services for the scheduler and shaper.

CPP queues are SP queues. The SPs of the packets are determined based on the time when they are added to a queue. Packets with a larger queue number have a higher priority.

4. Scheduler

The scheduler schedules packets based on SPs of queues. Packets in a high-priority queue are first scheduled.

Before being scheduled, the packets to be sent to the CPU are cached in queues. When being scheduled, the packets are sent to the CPU for processing.

Instruction

Only the SP scheduling policy is supported and cannot be modified.

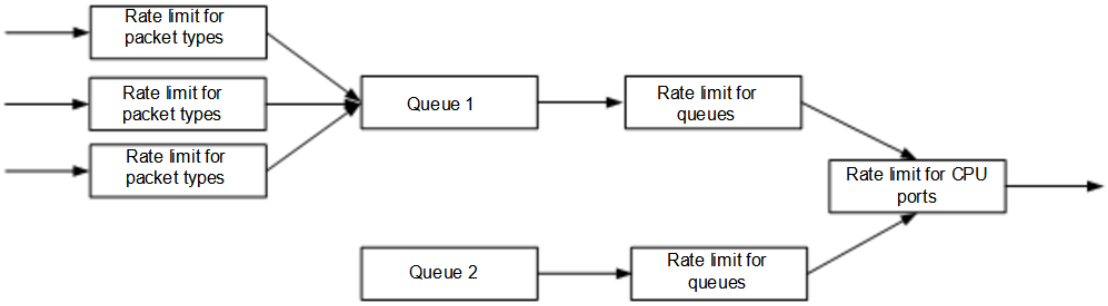
5. **Shaper**

The shaper shapes the packets to be sent to the CPU. When the actual rate of packets is greater than the shaping threshold, the packets must continue to cache in the queue and cannot be scheduled. When packet rates fluctuate, the shaper ensures that the rates of packets sent to the CPU are smooth (not more than the shaping threshold).

When the shaper is available, packets in a low-priority queue may be scheduled before those in a high-priority queue. If the rate of packets in a queue with a priority exceeds the shaping threshold, scheduling of the packets in this queue is stopped temporarily. Therefore, the shaper can prevent only the packets in high-priority queues from being scheduled.

Since the shaper limits the scheduling rates of packets, it provides the rate limit function. The shaper provides level-2 rate limit for priority queues and all packets sent to the CPU (CPU port). The shaper and meter functions provide 3-level rate limit and level-3 protection for the CPU.

Figure 1-1 3-Level Rate Limit of CPP



1.2 Configuration Task Summary

CPP configuration includes the following tasks:

- Configuring Basic Attributes of CPP
- Configuring Alarm Print Function of CPP
- Configuring Automatic Protection Function
- Configuring Automatic Protection Function for Ports

1.3 Configuring Basic Attributes of CPP

1.3.1 Overview

The implementation model of CPP includes five basic functional components: classifier, meter, queue, scheduler, and shaper. A device uses fixed classification policies for the classifier and SP scheduling algorithms for the scheduler.

In different network scenarios, users can configure personalized CPPs based on the meter, queue, and shaper to prevent packet attacks and network flapping from affecting normal operation of the CPU of a device.

- Configure a bandwidth for a specific packet type based on the meter to limit the packet transmission rate. Packets beyond the bandwidth are directly discarded.
- Configure a priority queue for a specific packet type based on the queue function. Packets in a high-priority queue are scheduled first.
- Configure a bandwidth for a CPU port and that for a specific priority queue to limit the packet transmission rate. Packets beyond the bandwidth are directly discarded.

1.3.2 Restrictions and Guidelines

- In one of the following situations, you must adjust the bandwidth for a specific packet type:
  - If a type of packets are discarded though they do not initiate attacks, you must increase the bandwidth for this packet type.
  - If a type of packets initiate attacks and cause CPU running exception, you must decrease the bandwidth for this packet type.
- In one of the following situations, you must adjust the priority queue for a specific packet type:
  - If a type of packets initiate attacks and cause exception in other packets in the same queue, you may put this type of packets into an unused queue.
  - A type of packets may not be discarded, and they are in the same queue with other types of packets in use. To prevent the bandwidth for other types of packets in the same queue from discarding packets, you can put this type of packets into a high-priority queue.
- In one of the following situations, you must adjust the bandwidth for a priority queue:
  - If the bandwidth for a type of packets increases, causing the available bandwidth for other packets in the same priority queue to decrease, you must appropriately increase the bandwidth for this priority queue.
  - If attack packets are put into a priority queue and no other packets in use stay in the same queue, you should decrease the bandwidth for this priority queue.
- When you configure a bandwidth for a specific packet type, note that decreasing the bandwidth may affect the normal service traffic of this packet type. To apply CPP based on users, use the NFPP function. For more information about NFPP configuration, see "Configuring Network Foundation Protection Policy".
- The bandwidth configuration for a packet type must match the bandwidth configuration for a priority queue. For the three levels of CPU protection, applying only one level of protection may cause negative effects. For example, if you increase the bandwidth for a packet type, you must also adjust the bandwidth for its priority queue. Otherwise, the other packets in the same priority queue may be affected.
- You are not advised to modify the bandwidth of the CPU port.

1.3.3 Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Configure the basic attributes of CPP, and configure at least one task.

- Configure a bandwidth for a packet type.

  cpu-protect type *packet-type* **bandwidth** *bandwidth-value*

  Each packet type has a rate threshold (bandwidth) by default, and this rate threshold may be modified, but may not be removed.

  The default bandwidth for packet types depends on the actual product version.

- Configure a priority queue for a packet type.

  cpu-protect type *packet-type* **traffic-class** *traffic-class-number*

  Each packet type has a priority queue by default.

  The default priority queues for packet types depend on the actual product version.

- Configure a bandwidth for a priority queue.

  cpu-protect traffic-class *traffic-class-number* **bandwidth** *bandwidth-value*

  A shaping threshold (bandwidth) is determined for each priority queue by default, and this shaping threshold may be modified, but may not be removed.

  The default bandwidth for priority queues depends on the actual product version.

- Configure a bandwidth for the CPU interface.

  cpu-protect cpu bandwidth *bandwidth-value*

  The default bandwidth for the CPU port depends on the actual product version.

1.4 Configuring Alarm Print Function of CPP

1.4.1 Overview

After the alarm print function of CPP is enabled, you can view the alarm logs of lost packets and detect packet attacks to the CPU of the device timely. For the normal service packets that are discarded falsely, you can adjust the CPP configuration to avoid subsequent packet loss.

The configuration effect of the alarm print function is as follows:

- After the alarm print function is enabled for a packet type, the corresponding alarm log is printed if packets of this type are lost.
- After the alarm print function is enabled for a priority queue, the alarm log is printed if packets in this queue are lost.
- After an alarm period is configured, a device periodically detects the lost packets sent to the CPU, and the alarm is printed only once for the packet loss in this specified period.

1.4.2 Restrictions and Guidelines

- You can configure the alarm print function with two methods. Select either of them to validate this function. Commands configured between the configuration methods produce the same function effect. If they are configured, the later configured command takes effect. To facilitate management, you are advised to adopt one configuration method.

1.4.3 Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Configure an alarm period of CPP. Please configure only one task.

- **cpu-protect alarm period** *period-value*
- **cpp-warn warn-period** *interval*

The default alarm period of CPP is **10** minutes.

(4) Enable the alarm print function for a packet type. Please configure only one task.

- **cpu-protect alarm type** *packet-type* **enable**
- **cpp-warn type** *packet-type* **warn**

The alarm print function for packets is not enabled by default.

(5) Enable the alarm print function for a queue. Please configure only one task.

- **cpu-protect alarm traffic-class** *traffic-class-number* **enable**
- **cpp-warn traffic-class** *traffic-class-number* **warn**

The alarm print function for priority queues is not enabled by default.

1.5 Configuring Automatic Protection Function

1.5.1 Overview

The automatic protection function allows a device to protect the application layer data based on sessions. The function ensures the normal operation of attacked services.

- After the automatic protection function is enabled based on packet types, the dynamic link protocol packets of the service module can get sufficient signaling to reduce protocol flapping when the packets are attacked.
- After it configures a domain for the customized security packet type, the function allows the matched security packets to get sufficient signaling to reduce protocol flapping when the packets are attacked.
- After it configures bandwidth for user trust packets, the function allows the trust packets to get sufficient signaling for forwarding, and limits the excessive bandwidth occupied by the packets.

1.5.2 Restrictions and Guidelines

- When the hardware capacity is insufficient, the whitelist entries configured by the Command Line Interface (CLI) or automatically delivered by services fail to be installed. A syslog is printed through the FP component on the device to indicate an entry capacity problem. Finally, the packet whitelist becomes invalid.
- You can enable the automatic protection function of protocol packets either in global configuration mode or for a packet type. If you do not configure the automatic protection function for a packet type, the global configuration of the automatic protection function prevails. If you configure the automatic protection function for a packet type, the configuration for this packet type prevails and the global configuration is not used.

1.5.3 Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Enable the automatic protection function for protocol packets.

cpu-protect auto-defend [**type** *packet-type*] **enable**

When protocol flapping occurs frequently for a packet type, you can attempt to enable the automatic protection function for this packet type to prevent these packets from being affected by other protocol packets.

The automatic protection function for the dynamic routing protocol packets is disabled by default.

(4) Configure a trust packet type.

cpu-protect auto-defend trusted-host type *packet-type* { **src-ip** *ipv4-addres* | **src-ip6** *ipv6-address* | **src-mac** *mac-address* }

Each service is configured with a default trust packet type by default.

(5) Configure a bandwidth for packets under automatic protection.

cpu-protect auto-defend type *packet-type* **bandwidth** *value*

The bandwidth for trust packets under automatic protection is **80%** of the global bandwidth by default.

1.6 Configuring Automatic Protection Function for Ports

1.6.1 Overview

If a packet attack occurs on a port, the attack packets sent to the CPU through this port occupy bandwidth. As a result, other protocol packets cannot be sent to the CPU for processing and thus services are interrupted.

The automatic protection function can defend against packet attacks to CPU ports. The implementation effect is as follows:

- After the automatic protection function for a port is enabled, a device can detect packet attacks on this port.
- After it configures an automatic protection attack threshold for a port that transmits protocol packets, this function allows the packets to have sufficient signaling for forwarding and limits the excessive bandwidth occupied by the packets.
- After it configures a blacklist to impose a rate limiting threshold for the ports in the blacklist, this function allows service packets to get sufficient bandwidth on other ports.
- After it configures a monitoring period for a port, this function limits the rates of attack packets on this port within the period. After the period, this function limits the rates of attack packets if the protocol packets received on this port remain in excess of the attack threshold. Otherwise, the limit is removed.
- After the attack alarming function is enabled on a port, syslogs are generated and Simple Network Management Protocol (SNMP) Trap messages are sent when attacks occur or are cleared. The alarm message is triggered when they occur or are cleared. The alarm is not given repeatedly during a monitoring period.

1.6.2 Restrictions and Guidelines

- When the default hardware capacity is insufficient, the blacklist entries fail to be installed. A syslog is printed through the FP component on the device to indicate an entry capacity problem. Finally, the rate limit by the hardware may become ineffective and must be reconfigured by software.
- To detect a specific network environment, you can enable the packet statistics function on a port and check attacks based on the packet statistics. If any, enable the automatic protection function on the port and configure an attack threshold and a rate limiting threshold based on the environmental configuration.
- The blacklist can take effect on a port only when the automatic protection function is enabled on this port.
- You can enable the automatic protection function in global configuration mode or for a port. If you do not configure the automatic protection function for a port, the global configuration of the automatic protection function prevails. If you configure the automatic protection function for a port, the configuration for this port prevails and the global configuration is not used.
- You can configure the packet penalty threshold for an attacked port in global configuration mode or for a specific packet type. If you do not configure the penalty threshold for a specific packet type, you impose a penalty threshold on packets in the global configuration mode when an attack is detected on a port. If you configure a penalty threshold for a specific packet type, the configuration for this packet type prevails.

1.6.3 Procedure

(1) Enter the privileged EXEC mode.

enable

(2) Enter the global configuration mode.

configure terminal

(3) Enable the automatic protection function for a port.

cpu-protect auto-port-defend [**interface** *interface-type interface-number* | **interface range** *interface-type start-number end-number*] **enable**

The automatic protection function for a port is disabled by default.

(4) Configure an attack threshold for a packet type under automatic protection on a port.

cpu-protect auto-port-defend type *packet-type* **attack-threshold** *threshold-value*

No attack threshold is configured for a packet type under automatic protection on a port by default.

(5) Configure a rate limiting threshold for all packets on an attacked port.

cpu-protect auto-port-defend limit threshold *threshold-value*

In the monitoring period, a rate limiting threshold will be applied to a port if an attack is detected on the port.

The rate limiting threshold for all packets on an attacked port is **5%** of the bandwidth for this interface by default.

(6) Configure a rate limiting threshold for specific packets on an attacked port.

cpu-protect auto-port-defend limit type *packet-type* **threshold** *threshold-value*

In the monitoring period, a rate limiting threshold is applied to the specific packets on a port if an attack is detected on the port.

The rate limiting threshold for specific packets on an attacked port is **5%** of the bandwidth for the port packets by default.

(7) Configure a blacklist for a port.

cpu-protect auto-port-defend blacklist interface *interface-type interface-number*

If a port that normally sends and receives packets is not identified as an attack interface, but affects traffic of services, you can add this port to a blacklist to reduce the traffic of this port.

A port is a trust port and is not in the blacklist by default.

(8) Configure a monitoring period for a port.

cpu-protect auto-port-defend monitor-period *value*

The default attack monitoring period of a port is **300** seconds.

(9) Enable the attack alarm function for a port.

cpu-protect auto-port-defend alarm enable

The attack alarm function for a port is disabled by default. After an attack occurs or is cleared, syslogs are not output and SNMP Trap messages are not sent to prompt users.

1.7 Monitoring

Run the **show** command to check the running status of a configured function to verify the configuration effect.

Run the **clear** commands to clear information.

Notice

- Running the **clear** command may lose vital information and thus interrupt services.
- The output debugging information occupies system resources. Therefore, disable the debugging function immediately after use.

Table 1-1 Monitoring

Command	Purpose
clear cpu-protect counters [**device** *device-number*] [**slot** *slot-number*]	Clears CPP statistics.
clear cpu-protect counters mboard	Clears the CPP statistics on the master device.
clear cpu-protect statistics [**interface** *interface-type interface-number*]	Clears the CPP statistics on an interface.
show cpu-protect type *packet-type* [**device** *device-number*] [**slot** *slot-number*]	Displays the configuration and statistics of a packet type.
show cpu-protect traffic-class *traffic-class-number* [**device** *device-number*] [**slot** *slot-number*]	Displays the configuration and statistics of a priority queue.
show cpu-protect cpu	Displays the configuration on the CPU port.
show cpu-protect { **mboard** | **summary** }	Displays all the configuration and statistics on the

Command	Purpose
	master device.
show cpu-protect [**device** *device-number*] [**slot** *slot-number*]	Displays all the configuration and statistics of CPP.
show cpu-protect statistics [**interface** *interface-type interface-number*]	Displays the statistics on a CPP port.
show cpu-protect statistics type *packet-type*	Displays the statistics of a packet type on a CPP port.
show cpu-protect hardware-statistics [**device** *device-number*] [**slot** *slot-number*]	Displays the hardware statistics of a CPP packet type.
show cpu-protect auto-defend	Displays automatic protection configuration and whitelist packet statistics.
show cpu-protect auto-defend summary	Displays the automatic protection configuration.
show cpu-protect auto-defend config	Displays the trust packet configuration.
show cpu-protect auto-port-defend	Displays the blacklist configuration on a port.
show cpu-protect auto-port-defend statistics	Displays the blacklist statistics on a port.
show cpu-protect auto-port-defend summary	Displays the automatic protection configuration on a port.
show cpu-protect auto-port-defend attacked summary	Displays the packet attack information on a port.
show cpu-protect linkage summary	Displays the protocol correlation state.

Instruction

The above monitoring command is compatible with chassis devices and cassette devices, as well as the standalone mode and VSU mode.

If the **device** and **slot** keywords are not entered, the **clear** command is used to clear the statistics of all nodes and interfaces in the system and the **show** command is used to display the configuration on the master device.

In the stand-alone mode, the **device** keyword is invisible. For chassis devices, the **slot** keyword is used to specify a line card; for cassette devices, the **slot** keyword is invisible.

In the VSU mode, the **device** keyword specifies a chassis or cassette device. If the **device** keyword is not entered, it specifies the master chassis or the master device. For chassis devices, if the **device** keyword is entered, the **slot** keyword must also be entered to mark a line card in a chassis; for cassette devices, the **slot** keyword is invisible.

1.8 Configuration Examples

1.8.1 Configuring Basic Functions of CPP

1. **Requirements**

Address Resolution Protocol (ARP), IP, Open Shortest Path First (OSPF), dot1x, Virtual Router Redundancy Protocol (VRRP), Telnet and ICMP streams are available in the system. In the current configuration, ARP and dot1x streams are in priority queue 2; IP, ICMP and Telnet streams are in priority queue 4; OSPF streams are in priority queue 3; and VRRP streams are in priority queue 6. The bandwidth for each packet type is 10,000 pps; the bandwidth for each priority queue is 20,000 pps; and the bandwidth for the CPU port is 100,000 pps. If the device suffers ARP packet attacks, you must enable the CPP function to prevent the packet attacks and network flapping and ensure the normal running of other services on the device.

2. **Notes**

- Classify ARP attack packets into priority queue 1 and limit the bandwidth for ARP packets or the priority queue.
- Classify OSPF packets into priority queue 5.
- Classify IP attack packets (without the ARP entries) into priority queue 3 and limit the bandwidth for IP packets or the priority queue.

3. **Procedure**

Configure the basic attributes of CPP.

```
DeviceA> enable
DeviceA# configure terminal
DeviceA(config)# cpu-protect type arp traffic-class 1
DeviceA(config)# cpu-protect type arp bandwidth 5000
DeviceA(config)# cpu-protect type ospf traffic-class 5
DeviceA(config)# cpu-protect type v4uc-route traffic-class 3
DeviceA(config)# cpu-protect traffic-class 3 bandwidth 5000
DeviceA(config)# end
```

4. **Verification**

Run the **show cpu-protect** command to view the configuration and statistics.

```
DeviceA# show cpu-protect
%cpu port bandwidth: 100000(pps)
Traffic-class   Bandwidth(pps)  Rate(pps)  Drop(pps)
-------------   --------------  ---------  ---------
 0              20000           0          0
 1              20000           0          0
 2              20000           0          0
 3              5000            0          0
 4              20000           0          0
 5              20000           0          0
 6              20000           0          0
 7              20000           0          0
```

```
Packet Type         Traffic-class  Bandwidth(pps)  Rate(pps)  Drop(pps)  Total
Total Drop
------------------  -------------  --------------  ---------  ---------
---------     ----------
bpdu                6              128             0          0          0
0
arp                 1              5000            0          0          0
0
tpp                 6              128             0          0          0
0
dot1x               2              1500            0          0          0
0
gvrp                5              128             0          0          0
0
rldp                5              128             0          0          0
0
lacp                5              256             0          0          0
0
rerp                5              128             0          0          0
0
reup                5              128             0          0          0
0
lldp                5              768             0          0          0
0
cdp                 5              768             0          0          0
0
dhcps               2              1500            0          0          0
0
dhcps6              2              1500            0          0          0
0
dhcp6-client        2              1500            0          0          0
0
dhcp6-server        2              1500            0          0          0
0
dhcp-relay-c        2              1500            0          0          0
0
dhcp-relay-s        2              1500            0          0          0
0
option82            2              1500            0          0          0
0
tunnel-bpdu         2              128             0          0          0
0
tunnel-gvrp         2              128             0          0          0
0
unknown-v6mc        1              128             0          0          0
0
```

```
xgv6-ipmc           1               128             0               0               0
0
stargv6-ipmc        1               128             0               0               0
0
unknown-v4mc        1               128             0               0               0
0
xgv-ipmc            2               128             0               0               0
0
stargv-ipmc         2               128             0               0               0
0
udp-helper          1               128             0               0               0
0
dvmrp               4               128             0               0               0
0
igmp                2               1000            0               0               0
0
icmp                3               1600            0               0               0
0
ospf                5               2000            0               0               0
0
ospf3               5               2000            0               0               0
0
pim                 4               1000            0               0               0
0
pimv6               4               1000            0               0               0
0
rip                 4               128             0               0               0
0
ripng               4               128             0               0               0
0
vrrp                6               256             0               0               0
0
vrrpv6              6               256             0               0               0
0
ttl0                0               128             0               0               0
0
ttl1                0               2000            0               0               0
0
hop-limit           0               800             0               0               0
0
local-ipv4          3               4000            0               0               0
0
local-ipv6          3               4000            0               0               0
0
v4uc-route          3               800             0               0               0
0
```

```
v6uc-route          1               800             0           0           0
0
rt-host             4               3000            0           0           0
0
mld                 2               1000            0           0           0
0
nd-snp-ns-na        1               3000            0           0           0
0
nd-snp-rs           1               1000            0           0           0
0
nd-snp-ra-redirect  1               1000            0           0           0
0
erps                5               128             0           0           0
0
mpls-ttl0           4               128             0           0           0
0
mpls-ttl1           4               128             0           0           0
0
mpls-ctrl           4               128             0           0           0
0
isis                4               2000            0           0           0
0
bgp                 4               2000            0           0           0
0
cfm                 5               512             0           0           0
0
web-auth            2               2000            0           0           0
0
fcoe-fip            4               1000            0           0           0
0
fcoe-local          4               1000            0           0           0
0
bfd                 6               5120            0           0           0
0
micro-bfd           6               5120            0           0           0
0
micro-bfd-v6        6               5120            0           0           0
0
dldp                6               3200            0           0           0
0
other               0               4096            0           0           0
0
trill               4               1000            0           0           0
0
efm                 5               1000            0           0           0
0
```

```
ipv6-all            0               2000            0           0           0
0
ip-option           0               800             0           0           0
0
mgmt                -               4000            4           0           4639
0
dns                 2               200             0           0           0
0
sdn                 0               5000            0           0           0
0
sdn_of_fetch        0               5000            0           0           0
0
sdn_of_copy         0               5000            0           0           0
0
sdn_of_trap         0               5000            0           0           0
0
vxlan-non-uc        1               512             0           0           0
0
local-telnet        3               1000            0           0           0
0
local-snmp          3               1000            0           0           0
0
local-ssh           3               1000            0           0           0
0
```