Configuring Control Plane Policing on C9500X-28C8D model

- Restrictions for Control Plane Policing, on page 1
- Information About Control Plane Policing, on page 2
- How to Configure CoPP, on page 10
- Configuration Examples for Control Plane Policing, on page 13
- Monitoring CoPP, on page 14
- Feature History for Control Plane Policing, on page 14

Restrictions for Control Plane Policing

Restrictions for control plane policing (CoPP) include the following:

- Only ingress CoPP is supported. The **system-cpp-policy** policy-map is available on the control plane interface, and only in the ingress direction.
- Only the **system-cpp-policy** policy-map can be installed on the control plane interface.
- The **system-cpp-policy** policy-map and the system-defined classes cannot be modified or deleted.
- Only the **police** action is allowed under the **system-cpp-policy** policy-map. The police rate for system-defined classes must be configured only in packets per second (pps).
- We recommend not disabling the policer for a system-defined class map, that is, do not configure **no police rate** *rate* **pps** command. Doing so affects the overall system health in case of high traffic towards the CPU. Further, even if you disable the policer rate for a system-defined class map, the systems automatically reverts to the default policer rate after system bootup in order to protect the system bring-up process.
- The **show run** command does not display information about classes configured under `system-cpp policy`, when they are left at default values. Use the **show policy-map system-cpp-policy** or the **show policy-map control-plane** commands instead.

  You can continue use the **show run** command to display information about custom policies.
- A protocol with a large number of CPU-bound packets may impact other protocols in the same class, as some of these protocols share the same policer. For example, Address Resolution Protocol (ARP) shares 4000 hardware policers with an array of host protocols like Telnet, Internet Control Message Protocol (ICMP), SSH, FTP, and SNMP in the system-cpp-police-forus class. If there is an ARP poisoning or an

ICMP attack, hardware policers start throttling any incoming traffic that exceeds 4000 packets per second to protect the CPU and the overall integrity of the system. As a result, ARP and ICMP host protocols are dropped, along with any other host protocols that share the same class.

- The creation of user-defined class-maps is not supported.

Information About Control Plane Policing

This chapter describes how CoPP works on your device and how to configure it.

Overview of Control Plane Policing

The CoPP feature improves security on your device by protecting the CPU from unnecessary traffic and denial of service (DoS) attacks. It can also protect control traffic and management traffic from traffic drops caused by high volumes of other, lower priority traffic.

Your device is typically segmented into three planes of operation, each with its own objective:

- The data plane, to forward data packets.
- The control plane, to route data correctly.
- The management plane, to manage network elements.

You can use CoPP to protect most of the CPU-bound traffic and ensure routing stability, reachability, and packet delivery. Most importantly, you can use CoPP to protect the CPU from a DoS attack.

CoPP uses the modular QoS command-line interface (MQC) and CPU queues to achieve these objectives. Different types of control plane traffic are grouped together based on certain criteria, and assigned to a CPU queue. You can manage these CPU queues by configuring dedicated policers in hardware. For example, you can modify the policer rate for certain CPU queues (traffic-type), or you can disable the policer for a certain type of traffic.

Although the policers are configured in hardware, CoPP does not affect CPU performance or the performance of the data plane. But since it limits the number of packets going to CPU, the CPU load is controlled. This means that services waiting for packets from hardware may see a more controlled rate of incoming packets (the rate being user-configurable).

System-Defined Aspects of Control Plane Policing

When you power-up the device for the first time, the system automatically performs the following tasks:

- Looks for policy-map **system-cpp-policy**. If not found, the system creates and installs it on the control-plane.
- Creates 56 class-maps under **system-cpp-policy**.
- Enables all CPU meters by default, with their respective default rate.

The **system-cpp-policy** policy map is a system-default policy map, and normally, you do not have to expressly save it to the startup configuration of the device. But, a *failed* bulk synchronization with a standby device can result in the configuration being erased from the startup configuration. In case this happens, you have to

manually save the **system-cpp-policy** policy map to the startup configuration. Use the **show running-config** privileged EXEC command to verify that it has been saved:

```
policy-map system-cpp-policy
```

The following table (System-Defined Values for CoPP) lists the class-maps that the system creates when you load the device and the feature(s) associated with each class-maps.

Note The class maps, detailed in the following table, are based on the output of the **show policy-map system-cpp-policy** command executed on a switch running with the Cisco IOS XE 17.18.x software release. If you are using a different software release, execute the **show policy-map system-cpp-policy** command, on your device, to verify the available class maps.

Table 1: System-Defined Values for CoPP

Class Maps Names	Associated Feature(s)
system-cpp-default	• ACL Drop (ETH) • GOLD • MC snoop lookup miss • IPv4 Checksum • IPv4 Unknown • ACL Drop • ACL Log • RPF (unicast) • Glean-SUBNET
system-cpp-default-v4	• WCCP IPv4 • RSVP IPv4
system-cpp-default-v6	• WCCP IPv6 • RSVP IPv6
system-cpp-police-arp	ARP
system-cpp-police-icmp-v4	ICMP IPv4
system-cpp-police-icmp-v6	• ICMP IPv6 MC • ICMP IPv6 UC
system-cpp-police-isis	• ISIS (L2) • ISIS (L3)

Class Maps Names	Associated Feature(s)
system-cpp-police-ospf-v4	• OSPFv2 MC • OSPFv2 UC
system-cpp-police-ospf-v6	• OSPFv3 MC • OSPFv3 UC
system-cpp-police-bgp-v4	BGP IPv4
system-cpp-police-bgp-v6	BGP IPv6
system-cpp-police-ldp-v4	• LDP UDP SRC IPv4 • LDP UDP DST IPv4 • LDP TCP SRC IPv4 • LDP TCP DST IPv4
system-cpp-police-ldp-v6	• LDP UDP SRC IPv6 • LDP UDP DST IPv6 • LDP TCP SRC IPv6 • LDP UDP DST IPv6
system-cpp-police-rip-bc	RIP
system-cpp-police-rip-v4	RIP IPv4
system-cpp-police-rip-v6	RIP IPv6
system-cpp-police-igrp-v4	EIGRP IPv4
system-cpp-police-igrp-v6	EIGRP IPv6
system-cpp-police-fhrp-v4	• VRRP IPv4 • HSRP IPv4 • GLBP IPv4
system-cpp-police-fhrp-v6	• VRRP IPv6 • HSRP IPv6 • GLBP IPv6

Class Maps Names	Associated Feature(s)
system-cpp-police-bfd-v4	• BFD IPv4 • BFD ECHO IPv4 • BFD MHOP IPv4
system-cpp-police-bfd-v6	• BFD IPv6 • BFD ECHO IPv6 • BFD MHOP IPv6
system-cpp-police-mpls-oam	MPLS OAM IPv4
system-cpp-police-mcast-ctrl-v4	• PIM IPv4 • IPv4 MC
system-cpp-police-mcast-ctrl-v6	• PIM IPv6 • IPv6 MC
system-cpp-police-l2-control	• Cisco Protocols • LLDP • MACSEC
system-cpp-police-mac-learning	MAC learning
system-cpp-police-system-critical	System critical messages like heartbeat messages
system-cpp-police-topology-control	STP (IEEE)
system-cpp-police-proto-snoop-v4	• IGMP Enable • ARP Snooping • IPv4 IGMP
system-cpp-police-proto-snoop-v6	• MDNS • SISF • IPv6 Options
system-cpp-police-dhcp-snooping	• Egress DHCP Snooping • DHCP Snooping
system-cpp-police-forus	• Ping • SSH

Class Maps Names	Associated Feature(s)
system-cpp-police-lacp	LACP
system-cpp-police-mtu-ttl-fail	• TTL1 (MPLS) • TTL0 (MPLS) • MPLS-VPN-TTL1 • TTL1 (L3) • MTU (L3)
system-cpp-police-meraki-nt-v4	IPv4 cloud management traffic
system-cpp-police-meraki-nt-v6	IPv6 cloud management traffic
system-cpp-police-tcp-v4	• DHCPv4 Server TCP Bulk Query • TCP default IPv4
system-cpp-police-tcp-v6	• DHCPv6 Server TCP Bulk Query • TCP default IPv6
system-cpp-police-dhcp-v4	• DHCP Client (v4) • DHCP Server (v4) • DHCP(mirror)
system-cpp-police-dhcp-v6	• DHCP Client (v6) • DHCP Server (v6)
system-cpp-police-udp-v4	• DHCPv4 S to S • UDP default IPv4
system-cpp-police-udp-v6	• DHCPv6 S to S • UDP default IPv6
system-cpp-police-mcast-rpf-fail	• MC PIM RPF FAIL • MC SNOOP RPF FAIL • MC PIM RPF

Class Maps Names	Associated Feature(s)
system-cpp-police-mcast-data	• MC FWD Disabled (v4) • MC FWD Disabled (v6) • MC G PIM Punt • MC direct connect • MC SG PIM Punt • MC Snoop and direct connect • MC not found
system-cpp-police-mcast-register-v4	IPv4 PIM Reg Mcast
system-cpp-police-mcast-register-v6	IPv6 PIM Reg Mcast
system-cpp-police-sw-forward	• ETH HOP-OPT • IP Options(v4) • IPv6 DEST-OPT • IPv6 HOP-OPT • GLEAN • PBR GLEAN
system-cpp-police-fragment-v4	IPv4 fragmentation
system-cpp-police-fragment-v6	IPv6 fragment
system-cpp-police-unknown-uc	Unknown UC
system-cpp-police-lisp-ctrl	• UDP SRC PORT LISP IPv4 • UDP DST PORT LISP IPv4 • TCP SRC PORT LISP IPv4 • TCP DST PORT LISP IPv4
system-cpp-police-alt-sw-forward	IPv6 UC
system-cpp-police-svl-ipc	SVL IPC traffic
system-cpp-police-svl-ctrl	SVL control traffic
system-cpp-police-mdns	Multicast DNS traffic

User-Configurable Aspects of Control Plane Policing

You can perform these tasks to manage control plane traffic:

Note All `system-cpp-policy` configurations must be saved so they are retained after reboot.

Change the Policer Rate

You can do this by configuring a policer rate action (in packets per second), under the corresponding class-map, within the `system-cpp-policy` policy-map.

Set Policer Rates to Default

Set the policer for CPU queues to their default values, by entering the **cpp system-default** command in global configuration mode.

User Actions for an Upgrade

User actions for an upgrade – depending on upgrade method:

Upgrade Method	Condition	Action Time and Action	Purpose
Regular[1]	None	**After upgrade (required)** Enter the **cpp system-default** command in global configuration mode	To get the latest, default policer rates.

[1] Refers to a software upgrade method that involves a reload of the switch. Can be install or bundle mode.

Upgrading or Downgrading the Software Version

Software Version Upgrades and CoPP

When you upgrade the software version on your device, the system checks and make the necessary updates as required for CoPP (For instance, it checks for the `system-cpp-policy` policy map and creates it if missing). You may also have to complete certain tasks before or after the upgrade activity. This is to ensure that any configuration updates are reflected correctly and CoPP continues to work as expected. Depending on the method you use to upgrade the software, upgrade-related tasks may be optional or recommended in some scenarios, and mandatory in others.

The system actions and user actions for an upgrade, are described here. Also included, are any release-specfic caveats.

System Actions for an Upgrade

When you upgrade the software version on your device, the system performs these actions. This applies to all upgrade methods:

- If the device did not have a `system-cpp-policy` policy map before upgrade, then on upgrade, the system creates a default policy map.
- If the device had a `system-cpp-policy` policy map before upgrade, then on upgrade, the system does not re-generate the policy.

User Actions for an Upgrade

User actions for an upgrade – depending on upgrade method:

Upgrade Method	Condition	Action Time and Action	Purpose
Regular[2]	None	**After upgrade (required)** Enter the **cpp system-default** command in global configuration mode	To get the latest, default policer rates.
In-Service Software Upgrade (ISSU)[3]	If there are user-defined classes in the existing software version or If there are system-defined classes in the existing software version that are deprecated in a later release (for example: system-cpp-police-control-low-priority).	**Before and after upgrade (required)** Enter the **cpp system-default** command in global configuration mode	Enter the command before upgrade, to ensure that any required system configuration is updated, ensuring smooth ISSU operation. Enter the command after upgrade for the latest, default policer rates.

[2] Refers to a software upgrade method that involves a reload of the switch. Can be install or bundle mode.
[3] ISSU is supported only from one extended maintenance release to another. For more information, see In-Service Software Upgrade (ISSU).

Software Version Downgrades and CoPP

The system actions and user actions for a downgrade, are described here.

System Actions for a Downgrade

When you downgrade the software version on your device, the system performs these actions. This applies to all downgrade methods:

- The system retains the `system-cpp-policy` policy map on the device, and installs it on the control plane.

User Actions for a Downgrade

User actions for a downgrade:

Upgrade Method	Condition	Action Time and Action	Purpose
Regular[4]	None	No action required	Not applicable

Upgrade Method	Condition	Action Time and Action	Purpose
In-Service Software Upgrade (ISSU)[5]	None	No action required	Not applicable

[4] Refers to a software upgrade method that involves a reload of the switch. Can be install or bundle mode.
[5] ISSU downgrades are not supported.

If you downgrade the software version and then upgrade, the system action and user actions that apply are the same as those mentioned for upgrades.

How to Configure CoPP

Enabling a CPU Queue and Changing the Policer Rate

The procedure to enable a CPU queue and change the policer rate of a CPU queue is the same. Follow these steps:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	**policy-map** *policy-map-name*
	Example:
	`Device(config)# policy-map system-cpp-policy`
	`Device(config-pmap)#`
		Enters the policy map configuration mode.

Step 4	**class** *class-name*
	Example:
	`Device(config-pmap)# class system-cpp-police-protocol-snooping`
	`Device(config-pmap-c)#`
		Enters the class action configuration mode. Enter the name of the class that corresponds to the CPU queue you want to enable. See table *System-Defined Values for CoPP*.

	Command or Action	Purpose

Step 5	**police rate** *rate* **pps**
	Example:
	`Device(config-pmap-c)# police rate 100 pps`
	`Device(config-pmap-c-police)#`
		Specifies an upper limit on the number of incoming packets processed per second, for the specified traffic class.
		Note The rate you specify is applied to all CPU queues that belong to the class-map you have specified.

Step 6	exit
	Example:
	`Device(config-pmap-c-police)# exit`
	`Device(config-pmap-c)# exit`
	`Device(config-pmap)# exit`
	`Device(config)#`
		Returns to the global configuration mode.

Step 7	**control-plane**
	Example:
	`Device(config)# control-plane`
	`Device(config-cp)#`
		Enters the control plane (config-cp) configuration mode

Step 8	**service-policy input** *policy-name*
	Example:
	`Device(config)# control-plane`
	`Device(config-cp)#service-policy input system-cpp-policy`
	`Device(config-cp)#`
		Installs system-cpp-policy in FED. This command is required for you to see the FED policy. Not configuring this command will lead to an error.

Step 9	**end** **Example:** `Device(config-cp)# end`	Returns to the privileged EXEC mode.
Step 10	**show policy-map control-plane** **Example:** `Device# show policy-map control-plane`	Displays all the classes configured under `system-cpp policy`, the rates configured for the various traffic types, and statistics

(Optional) Disabling a CPU Queue

Follow these steps to disable a CPU queue:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	**policy-map** *policy-map-name*
	Example:
	`Device(config)# policy-map system-cpp-policy`
	`Device(config-pmap)#`
		Enters the policy map configuration mode.

Step 4	**class** *class-name*
	Example:
	`Device(config-pmap)# class system-cpp-police-protocol-snooping`
	`Device(config-pmap-c)#`
		Enters the class action configuration mode. Enter the name of the class that corresponds to the CPU queue you want to disable. See the table, *System-Defined Values for CoPP*.

Step 5	**no police rate** *rate* **pps** **Example:** `Device(config-pmap-c)# no police rate 100 pps`	Disables incoming packet processing for the specified traffic class. **Note** This disables all CPU queues that belong to the class-map you have specified.
Step 6	**end** **Example:** `Device(config-pmap-c)# end`	Returns to the privileged EXEC mode.
Step 7	**show policy-map control-plane** **Example:** `Device# show policy-map control-plane`	Displays all the classes configured under `system-cpp policy` and the rates configured for the various traffic types and statistics.

Setting the Default Policer Rates for All CPU Queues

Follow these steps to set the policer rates for all CPU queues to their default rates:

Procedure

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

Step 3	**cpp system-default**
	Example:
	`Device(config)# cpp system-default`
	`Defaulting CPP : Policer rate for all classes will be set to their defaults`
		Sets the policer rates for all the classes to the default rate.

Step 4	**end** **Example:** `Device(config)# end`	Returns to the privileged EXEC mode.

Configuration Examples for Control Plane Policing

Example: Enabling and Changing the Policer Rate of a CPU Queue

This example shows how to enable a CPU queue or to change the policer rate of a CPU queue. Here the **system-cpp-police-proto-snoop-v4** CPU queue is enabled with the policer rate of **2000 pps** .

```
Device> enable
Device# configure terminal
Device(config)# policy-map system-cpp-policy
Device(config-pmap)# class system-cpp-police-proto-snoop-v4
Device(config-pmap-c)# police rate 2000 pps
Device(config-pmap-c-police)# end
```

```
Device# show policy-map system-cpp-policy | sec system-cpp-police-proto-snoop-v4
   Class system-cpp-police-proto-snoop-v4
     police rate 2000 pps
       conform-action transmit
       exceed-action drop
```

Example: Disabling a CPU Queue

This example shows how to disable a CPU queue. Here the **class system-cpp-police-proto-snoop-v4** CPU queue is disabled.

```
Device> enable
Device# configure terminal
Device(config)# policy-map system-cpp-policy
Device(config-pmap)# class system-cpp-police-proto-snoop-v4
Device(config-pmap-c)# no police rate 100 pps
Device(config-pmap-c)# end
```

Example: Setting the Default Policer Rates for All CPU Queues

This example shows how to set the policer rates for all CPU queues to their default and then verify the setting.

Note For some CPU queues, the `default rate` and the `set rate` values will not be the same, even if you set the default rate for all classes. This because the set rate is rounded off to the nearest multiple of 200. This behavior is controlled by the clock speed of your device. In the sample output below, the default and set rate values for `DHCP Snooping` and `NFL SAMPLED DATA` display this difference.

```
Device> enable
Device# configure terminal
Device(config)# cpp system-default
Defaulting CPP : Policer rate for all classes will be set to their defaults
Device(config)# end
```

Monitoring CoPP

Use these commands to display policer settings, such as, traffic types and policer rates (user-configured and default rates) for CPU queues:

Command	Purpose
show policy-map control-plane	Displays the rates configured for the various traffic types
show policy-map system-cpp-policy	Displays all the classes configured under system-cpp policy, and policer rates
show platform software fed switch active punt entries	Displays the policies,the priorities, and the rates configured for the various traffic types

Feature History for Control Plane Policing

This table provides release and related information for the features explained in this module.

These features are available in all the releases subsequent to the one they were introduced in, unless noted otherwise.

Release	Feature	Feature Information
Cisco IOS XE Cupertino 17.7.1	Control Plane Policing (CoPP) or CPP	The CoPP feature improves security on your device by protecting the CPU from unnecessary traffic, or DoS traffic, and by prioritizing control plane and management traffic.

Use the Cisco Feature Navigator to find information about platform and software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com.