



# Configuring Control Plane Policing

- [Restrictions for CoPP, on page 1](#)
- [Information About CoPP, on page 2](#)
- [How to Configure CoPP, on page 6](#)
- [Configuration Examples for CoPP, on page 10](#)
- [Monitoring CoPP, on page 13](#)
- [Feature Information for CoPP, on page 14](#)

## Restrictions for CoPP

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)
- One or more CPU queues are part of each class-map. Where multiple CPU queues belong to one class-map, changing the policer rate of a class-map affects all CPU queues that belong to that class-map. Similarly, disabling the policer in a class-map disables all queues that belong to that class-map. See *Table: System-Defined Values for CoPP* for information about which CPU queues belong to each class-map.
- Disabling the policer for a system-defined class map is not recommended. That is, do not configure the **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.

- Starting from Cisco IOS XE Fuji 16.8.1a, the creation of user-defined class-maps is not supported.

# Information About CoPP

This chapter describes how control plane policing (CoPP) works on your device and how to configure it.

## CoPP Overview

The CoPP feature improves security on your device protecting the CPU from unnecessary traffic and DoS attacks. It can also protect control 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 CoPP

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 18 class-maps under **system-cpp-policy**.

The next time you power-up the device, the system detects the policy and class maps that have already been created.

- Enables all CPU queues by default, with their respective default rate. The default rates are indicated in the table System-Defined Values for CoPP.

The following table lists the class-maps that the system creates when you load the device. It lists the policer that corresponds to each class-map and one or more CPU queues that are grouped under each class-map. There is a one-to-one mapping of a class-map to a policer; and one-to-many mapping of a class-map to CPU queues.

Table 1: System-Defined Values for CoPP

Class Maps Names	Policer Index (Policer No.)	CPU queues (Queue No.)	Default Policer Rate (pps)
system-cpp- police-data	WK_CPP_POLICE_DATA(0)	WK_CPU_Q_ICMP_GEN(3)	600
		WK_CPU_Q_BROADCAST(12)	600
		WK_CPU_Q_ICMP_REDIRECT (6)	600
system-cpp-police-l2- control	WK_CPP_POLICE_L2_CONTROL(1)	WK_CPU_Q_L2_CONTROL(1)	2000
system-cpp-police-routing-control	WK_CPP_POLICE_ROUTING_CONTROL(0)	WK_CPU_Q_ROUTING_CONTROL(4)	5400
		WK_CPU_Q_LOW_LATENCY (27)	5400
system-cpp-police-punt-webauth	WK_CPP_POLICE_PUNT_WEBAUTH(7)	WK_CPU_Q_PUNT_WEBAUTH(22)	1000
system-cpp-police-topology-control	WK_CPP_POLICE_TOPOLOGY_CONTROL(8)	WK_CPU_Q_TOPOLOGY_CONTROL(15)	13000
system-cpp-police- multicast	WK_CPP_POLICE_MULTICAST(9)	WK_CPU_Q_TRANSIT_TRAFFIC(18)	500
		WK_CPU_Q_MCAST_DATA(30)	500
system-cpp-police-sys- data	WK_CPP_POLICE_SYS_DATA (10)	WK_CPU_Q_OPENFLOW (13)	100
		WK_CPU_Q_CRYPTO_CONTROL(23)	100
		WK_CPU_Q_EXCEPTION(24)	100
		WK_CPU_Q_EGR_EXCEPTION(28)	100
		WK_CPU_Q_NF_SAMPLED_DATA(26)	100
		WK_CPU_Q_GOLD_PKT(31)	100
WK_CPU_Q_RPF_FAILED(19)	100		
system-cpp-police-dot1x-auth	WK_CPP_POLICE_DOT1X(11)	WK_CPU_Q_DOT1X_AUTH(0)	1000
system-cpp-police-protocol-snooping	WK_CPP_POLICE_PROTO_SNOOPING	WK_CPU_Q_PROTO_SNOOPING(16)	2000
system-cpp-police-dhcp-snooping	WK_CPP_DHCP_SNOOPING	WK_CPU_Q_DHCP_SNOOPING(17)	500
system-cpp-police-sw-forward	WK_CPP_POLICE_SW_FWD (13)	WK_CPU_Q_SW_FORWARDING_Q(14)	1000
		WK_CPU_Q_LOGGING(21)	1000
		WK_CPU_Q_L2_LVX_DATA_PACK (11)	1000
system-cpp-police-forus	WK_CPP_POLICE_FORUS(14)	WK_CPU_Q_FORUS_ADDR_RESOLUTION(5)	4000
		WK_CPU_Q_FORUS_TRAFFIC(2)	4000

Class Maps Names	Policer Index (Policer No.)	CPU queues (Queue No.)	Default Policer Rate (pps)
system-cpp-police-multicast-end-station	WK_CPP_POLICE_MCAST_END_STATION(5)	WK_CPU_Q_MCAST_END_STATION_SERVICE(20)	2000
system-cpp-default	WK_CPP_DEFAULT(1)	WK_CPU_Q_INTER_FED_TRAFFIC WK_CPU_Q_EWLC_CONTROL(9) WK_CPU_Q_EWLC_DATA(10)	2000 2000 2000
system-cpp-police-stackwise-vit-control	WK_CPP_STACKWISE_VIRTUAL_CTRL(6)	WK_CPU_Q_STACKWISE_VIRTUAL_CTRL(29)	8000
system-cpp-police-l2lvx-control	WK_CPP_L2_LVX_CTRL_PACK	WK_CPU_Q_L2_LVX_CTRL_PACK(8)	1000
system-cpp-police-high-rate-app	WK_CPP_HIGH_RATE_APP	WK_CPU_Q_HIGH_RATE_APP	13000
system-cpp-police-system-critical	WK_CPP_SYSTEM_CRITICAL	WK_CPU_Q_SYSTEM_CRITICAL	1000

## User-Configurable Aspects of CoPP

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.

### Enable or Disable a Policer for CPU Queues

Enable a policer for a CPU queue, by configuring a policer action (in packets per second) under the corresponding class-map, within the `system-cpp-policy` policy-map.

Disable a policer for CPU queue, by removing the policer action under the corresponding class-map, within the `system-cpp-policy` policy-map.



### Note

If a default policer is already present, carefully consider and control its removal; otherwise the system may see a CPU hog or other anomalies, such as control packet drops.

### 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.

# 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-specific 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 <sup>1</sup>	None	<b>After upgrade (required)</b> Enter the <code>cpp system-default</code> command in global configuration mode	To get the latest, default policer rates.

<sup>1</sup> Refers to a software upgrade method that involves a reload of the switch. Can be install or bundle mode.

## 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 <sup>2</sup>	None	No action required	Not applicable

<sup>2</sup> Refers to a software upgrade method that involves a reload of the switch. Can be install or bundle mode.

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 or 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:

#### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class** *class-name*
5. **police rate** *rate* **pps**
6. **exit**
7. **control-plane**
8. **service-policy input** *policy-name*
9. **end**
10. **show policy-map control-plane**

#### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b>  <b>Example:</b>  Device> <b>enable</b>	Enables privileged EXEC mode.  • Enter your password if prompted.
<b>Step 2</b>	<b>configure terminal</b>  <b>Example:</b>  Device# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 3</b>	<b>policy-map</b> <i>policy-map-name</i>  <b>Example:</b>	Enters the policy map configuration mode.

	Command or Action	Purpose
	Device(config)# <b>policy-map system-cpp-policy</b> Device(config-pmap)#	
<b>Step 4</b>	<b>class class-name</b>  <b>Example:</b>  Device(config-pmap)# <b>class</b> <b>system-cpp-police-protocol-snooping</b> 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 <i>System-Defined Values for CoPP</i> .
<b>Step 5</b>	<b>police rate rate pps</b>  <b>Example:</b>  Device(config-pmap-c)# <b>police rate 100 pps</b> Device(config-pmap-c-police)#	Specifies an upper limit on the number of incoming packets processed per second, for the specified traffic class.  <b>Note</b> The rate you specify is applied to all CPU queues that belong to the class-map you have specified.
<b>Step 6</b>	<b>exit</b>  <b>Example:</b>  Device(config-pmap-c-police)# <b>exit</b> Device(config-pmap-c)# <b>exit</b> Device(config-pmap)# <b>exit</b> Device(config)#	Returns to the global configuration mode.
<b>Step 7</b>	<b>control-plane</b>  <b>Example:</b>  Device(config)# <b>control-plane</b> Device(config-cp)#	Enters the control plane (config-cp) configuration mode
<b>Step 8</b>	<b>service-policy input policy-name</b>  <b>Example:</b>  Device(config)# <b>control-plane</b> Device(config-cp)# <b>service-policy input</b> <b>system-cpp-policy</b> 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.
<b>Step 9</b>	<b>end</b>  <b>Example:</b>  Device(config-cp)# <b>end</b>	Returns to the privileged EXEC mode.
<b>Step 10</b>	<b>show policy-map control-plane</b>  <b>Example:</b>  Device# <b>show policy-map control-plane</b>	Displays all the classes configured under <code>system-cpp policy</code> , the rates configured for the various traffic types, and statistics

## Disabling a CPU Queue

Follow these steps to disable a CPU queue:

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class** *class-name*
5. **no police rate** *rate* **pps**
6. **end**
7. **show policy-map control-plane**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b> <b>Example:</b> Device> <b>enable</b>	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> Device# <b>configure terminal</b>	Enters global configuration mode.
<b>Step 3</b>	<b>policy-map</b> <i>policy-map-name</i> <b>Example:</b> Device(config)# <b>policy-map system-cpp-policy</b> Device(config-pmap)#	Enters the policy map configuration mode.
<b>Step 4</b>	<b>class</b> <i>class-name</i> <b>Example:</b> Device(config-pmap)# <b>class</b> <b>system-cpp-police-protocol-snooping</b> 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, <i>System-Defined Values for CoPP</i> .
<b>Step 5</b>	<b>no police rate</b> <i>rate</i> <b>pps</b> <b>Example:</b> Device(config-pmap-c)# <b>no police rate 100 pps</b>	Disables incoming packet processing for the specified traffic class.  <b>Note</b> This disables all CPU queues that belong to the class-map you have specified.



	Command or Action	Purpose
<b>Step 6</b>	<b>end</b> <b>Example:</b> <pre>Device(config-pmap-c) # end</pre>	Returns to the privileged EXEC mode.
<b>Step 7</b>	<b>show policy-map control-plane</b> <b>Example:</b> <pre>Device# show policy-map control-plane</pre>	Displays all the classes configured under <code>system-cpp policy</code> 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:

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **cpp system-default**
4. **end**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b> <b>Example:</b> <pre>Device&gt; enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
<b>Step 2</b>	<b>configure terminal</b> <b>Example:</b> <pre>Device# configure terminal</pre>	Enters global configuration mode.
<b>Step 3</b>	<b>cpp system-default</b> <b>Example:</b> <pre>Device(config)# cpp system-default Defaulting CPP : Policer rate for all classes will be set to their defaults</pre>	Sets the policer rates for all the classes to the default rate.
<b>Step 4</b>	<b>end</b> <b>Example:</b>	Returns to the privileged EXEC mode.

	Command or Action	Purpose
	Device(config)# <b>end</b>	

## Configuration Examples for CoPP

### Example: Enabling a CPU Queue or 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 **class system-cpp-police-protocol-snooping** 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-protocol-snooping
Device(config-pmap-c)# police rate 2000 pps
Device(config-pmap-c-police)# end
```

```
Device# show policy-map control-plane
Control Plane
```

```
Service-policy input: system-cpp-policy
```

```
<output truncated>
```

```
Class-map: system-cpp-police-dot1x-auth (match-any)
 0 packets, 0 bytes
 5 minute offered rate 0000 bps, drop rate 0000 bps
Match: none
police:
  rate 1000 pps, burst 244 packets
  conformed 0 bytes; actions:
    transmit
  exceeded 0 bytes; actions:
    drop
```

```
Class-map: system-cpp-police-protocol-snooping (match-any)
 0 packets, 0 bytes
 5 minute offered rate 0000 bps, drop rate 0000 bps
Match: none
police:
  rate 2000 pps, burst 488 packets
  conformed 0 bytes; actions:
    transmit
  exceeded 0 bytes; actions:
    drop
```

```
<output truncated>
```

```
Class-map: class-default (match-any)
 0 packets, 0 bytes
 5 minute offered rate 0000 bps, drop rate 0000 bps
```

```
Match: any
```

## Example: Disabling a CPU Queue

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

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

```
Device# show running-config | begin system-cpp-policy
```

```
policy-map system-cpp-policy
class system-cpp-police-data
  police rate 200 pps
class system-cpp-police-sys-data
  police rate 100 pps
class system-cpp-police-sw-forward
  police rate 1000 pps
class system-cpp-police-multicast
  police rate 500 pps
class system-cpp-police-multicast-end-station
  police rate 2000 pps
class system-cpp-police-punt-webauth
class system-cpp-police-l2-control
class system-cpp-police-routing-control
  police rate 500 pps
class system-cpp-police-control-low-priority
class system-cpp-police-wireless-priority1
class system-cpp-police-wireless-priority2
class system-cpp-police-wireless-priority3-4-5
class system-cpp-police-topology-control
class system-cpp-police-dot1x-auth
class system-cpp-police-protocol-snooping
class system-cpp-police-forus
class system-cpp-default
```

```
<output truncated>
```

## 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.

```
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
```

```
Device# show platform hardware fed switch 1 qos queue stats internal cpu policer
CPU Queue Statistics
```

```
=====
```

## Example: Setting the Default Policer Rates for All CPU Queues

QId	PlcIdx	Queue Name	Enabled	(default) Rate	(set) Rate	Queue Drop (Bytes)	Queue Drop (Frames)
0	11	DOT1X Auth	Yes	1000	1000	0	0
1	1	L2 Control	Yes	2000	2000	0	0
2	14	Forus traffic	Yes	4000	4000	0	0
3	0	ICMP GEN	Yes	600	600	0	0
4	2	Routing Control	Yes	5400	5400	0	0
5	14	Forus Address resolution	Yes	4000	4000	0	0
6	0	ICMP Redirect	Yes	600	600	0	0
7	16	Inter FED Traffic	Yes	2000	2000	0	0
8	4	L2 LVX Cont Pack	Yes	1000	1000	0	0
9	16	EWLC Control	Yes	2000	2000	0	0
10	16	EWLC Data	Yes	2000	2000	0	0
11	13	L2 LVX Data Pack	Yes	1000	1000	0	0
12	0	BROADCAST	Yes	600	600	0	0
13	10	Openflow	Yes	100	100	0	0
14	13	Sw forwarding	Yes	1000	1000	0	0
15	8	Topology Control	Yes	13000	13000	0	0
16	12	Proto Snooping	Yes	2000	2000	0	0
17	6	DHCP Snooping	Yes	500	500	0	0
18	9	Transit Traffic	Yes	500	500	0	0
19	10	RPF Failed	Yes	100	100	0	0
20	15	MCAST END STATION	Yes	2000	2000	0	0
21	13	LOGGING	Yes	1000	1000	0	0
22	7	Punt Webauth	Yes	1000	1000	0	0
23	18	High Rate App	Yes	13000	13000	0	0
24	10	Exception	Yes	100	100	0	0
25	3	System Critical	Yes	1000	1000	0	0
26	10	NFL SAMPLED DATA	Yes	100	200	0	0
27	2	Low Latency	Yes	5400	5400	0	0
28	10	EGR Exception	Yes	100	100	0	0
29	5	Stackwise Virtual OOB	Yes	8000	8000	0	0

30	9	MCAST Data	Yes	500	500	0	0
31	10	Gold Pkt	Yes	100	100	0	0

\* NOTE: CPU queue policer rates are configured to the closest hardware supported value

CPU Queue Policer Statistics

```

=====
Policer      Policer Accept  Policer Accept  Policer Drop  Policer Drop
  Index      Bytes          Frames          Bytes         Frames
-----
0            0              0              0             0
1            0              0              0             0
2            0              0              0             0
3            0              0              0             0
4            0              0              0             0
5            0              0              0             0
6            0              0              0             0
7            0              0              0             0
8            0              0              0             0
9            0              0              0             0
10           0              0              0             0
11           0              0              0             0
12           0              0              0             0
13           0              0              0             0
14           0              0              0             0
15           0              0              0             0
16           0              0              0             0
17           0              0              0             0
18           0              0              0             0
    
```

CPP Classes to queue map

```

=====
PlcIdx CPP Class                                     : Queues
-----
0      system-cpp-police-data                       : ICMP GEN/BROADCAST/ICMP Redirect/
10     system-cpp-police-sys-data                   : Openflow/Exception/EGR Exception/NFL
      SAMPLED DATA/Gold Pkt/RPF Failed/
13     system-cpp-police-sw-forward                 : Sw forwarding/LOGGING/L2 LVX Data Pack/
9      system-cpp-police-multicast                  : Transit Traffic/MCAST Data/
15     system-cpp-police-multicast-end-station      : MCAST END STATION /
7      system-cpp-police-punt-webauth               : Punt Webauth/
1      system-cpp-police-l2-control                 : L2 Control/
2      system-cpp-police-routing-control            : Routing Control/Low Latency/
3      system-cpp-police-system-critical            : System Critical/
4      system-cpp-police-l2lvx-control              : L2 LVX Cont Pack/
8      system-cpp-police-topology-control           : Topology Control/
11     system-cpp-police-dot1x-auth                 : DOT1X Auth/
12     system-cpp-police-protocol-snooping          : Proto Snooping/
6      system-cpp-police-dhcp-snooping              : DHCP Snooping/
14     system-cpp-police-forus                       : Forus Address resolution/Forus traffic/
5      system-cpp-police-stackwise-virt-control     : Stackwise Virtual OOB/
16     system-cpp-default                           : Inter FED Traffic/EWLC Control/EWLC Data/
18     system-cpp-police-high-rate-app              : High Rate App/
    
```

## 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
<b>show policy-map control-plane</b>	Displays the rates configured for the various traffic types
<b>show policy-map system-cpp-policy</b>	Displays all the classes configured under system-cpp policy, and policer rates
<b>show platform hardware fed switch {switch-number} qos que stats internal cpu policer</b>	Displays the rates configured for the various traffic types
<b>show platform software fed {switch-number} qos policy target status</b>	Displays information about policy status and the target port type.

## Feature Information for CoPP

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

Feature	Release	Feature Information
Control Plane Policing (CoPP) or CPP	Cisco IOS XE 3.3SE	This feature was introduced.
CLI configuration for CoPP	Cisco IOS XE Denali 16.1.2	This feature was made user-configurable. CLI configuration options to enable and disable CPU queues, to change the policer rate, and to set policer rates to default.
User-defined class maps	Cisco IOS XE Everest 16.5.1a	Starting with this release, you can create class maps (with filters) and add these user-defined class maps to system-cpp-policy.

Feature	Release	Feature Information
Changes in system-defined values for CoPP	Cisco IOS XE Everest 16.6.1	<p>These new system-defined classes were introduced:</p> <ul style="list-style-type: none"> <li>• system-cpp-police-stackwise-virt-control</li> <li>• system-cpp-police-l2lvs-control</li> </ul> <p>These new CPU queues were added to the existing system-cpp-default class:</p> <ul style="list-style-type: none"> <li>• WK_CPU_Q_UNUSED (7)</li> <li>• WK_CPU_Q_EWLC_CONTROL(9)</li> <li>• WK_CPU_Q_EWLC_DATA(10)</li> </ul> <p>This new CPU queues was added to the existing system-cpp-police-sw-forward: WK_CPU_Q_L2_LVX_DATA_PACK (11)</p> <p>This CPU queue is no longer available: WK_CPU_Q_SGT_CACHE_FULL(27)</p>
Removal of support for user-defined class-maps and changes in system-defined values for CoPP	Cisco IOS XE Fuji 16.8.1a	<ul style="list-style-type: none"> <li>• Starting from this release, the creation of user-defined class-maps is not supported.</li> <li>• This new system-defined class was introduced: system-cpp-police-dhcp-snooping</li> <li>• This new CPU queue was added to the existing system-cpp-default class: WK_CPU_Q_INTER_FED_TRAFFIC</li> <li>• These CPU queues are no longer available: <ul style="list-style-type: none"> <li>• WK_CPU_Q_SHOW_FORWARD</li> <li>• WK_CPU_Q_UNUSED</li> </ul> </li> <li>• The default policer rate (pps) for some CPU queues has changed: <ul style="list-style-type: none"> <li>• The default rate for WK_CPU_Q_EXCEPTION(24) was changed to 100</li> <li>• The default rate for all the CPU queues under system-cpp-default was increased to 2000.</li> <li>• The default rate for all the CPU queues under system-cpp-police-forus was increased to 4000.</li> </ul> </li> </ul>

Feature	Release	Feature Information
Changes in system-defined values for CoPP	Cisco IOS XE Fuji 16.9.1	<p>Starting with this release, eighteen system-defined classes are created under <code>system-cpp-policy</code>.</p> <p>These new system-defined classes were introduced:</p> <ul style="list-style-type: none"> <li>• <code>system-cpp-police-high-rate-app</code></li> <li>• <code>system-cpp-police-system-critical</code></li> </ul> <p>This was added to class <code>system-cpp-police-sys-data</code>: CPU queue <code>WK_CPU_Q_OPENFLOW (13)</code>.</p> <p>This CPU queue is no longer available: <code>WK_CPU_Q_LEARNING_CACHE_OVFL(13)</code>.</p>
Deprecation of system-defined class map	Cisco IOS XE Fuji 16.9.4	This system-defined class map was deprecated: <code>system-cpp-police-control-low-priority</code>