



Control Plane Policing

This document provides detailed understanding about Control Plane Policing and its configuration.

- [Feature history for CoPP, on page 1](#)
- [Control plane policing, on page 2](#)
- [System-defined aspects of CoPP, on page 2](#)
- [User-configurable aspects of CoPP, on page 11](#)
- [Software upgrades and CoPP, on page 11](#)
- [Software downgrades and CoPP, on page 12](#)
- [Restrictions for CoPP, on page 13](#)
- [Configure CoPP, on page 14](#)
- [Configuration examples, on page 17](#)
- [Monitor CoPP, on page 18](#)

Feature history for CoPP

This table provides release and platform support 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 name and description	Supported platform
Cisco IOS XE 26.2.1ea	Control Plane Policing (CoPP): CoPP feature support has been introduced.	Cisco C9550 Series Smart Switches
Cisco IOS XE 17.18.1	Control Plane Policing (CoPP): 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.	Cisco C9350 Series Smart Switches Cisco C9610 Series Smart Switches

Control plane policing

Control plane policing (CoPP) is a network security feature that

- improves security on your device by protecting the CPU from unnecessary traffic and denial of service (DoS) attacks
- protects control traffic and management traffic from traffic drops caused by high volumes of other, lower priority traffic, and
- uses the modular QoS command-line interface (MQC) and CPU queues to achieve these objectives.

Device operation planes

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.

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 these tasks:

- Looks for policy-map **system-cpp-policy**. If not found, the system creates and installs it on the control-plane.
- Creates 62 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
```

This 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-map.



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.
 - Each class map maps to a control plane policer index and a default meter rate. The CPU queue and port are determined independently by the application or trap policy.
-

Table 1: System-defined values for CoPP

Class-map names	Associated feature(s)
system-cpp-default	• ACL Drop (ETH) • ACL Drop (punt to cpu) • ACL LOG • GACL webauth punt (http/https punt to cpu) • GLEAN-SUBNET • GOLD • IPv4 Checksum • IPV4 UNKNOWN • L3 ACL Drop (no punt) • LISP-SMR • LPTS • MC snoop lookup miss • Meraki LSP • Mirror: ACL MIRROR • Mirror: EPC • Mirror: LISP-SMR-SIGNAL • Mirror: SGACL MIRROR • Mirror: SVL STDBY BD • Mirror: SVL STDBY MC • NRM Bridged (IPv4) • NRM Bridged (IPv6) • RACL Drop (punt to cpu for icmp generation) • RPF (unicast) • SDA HOST LEARNING • SDA SIGNAL-FWD • Term: GOLD
system-cpp-default-v4	• RSVP IPv4 • WCCP IPv4

Class-map names	Associated feature(s)
system-cpp-drop	• DHCPv4 C to S • DHCPv4 S to C • DHCPv6 C to S • DHCPv6 S to C • IEEE (FC) DROP • LPM Drop • Multicast SSDP • No Termination on L3 Port • Term: IEEE (FC)
system-cpp-police-arp	• ARP • Mirror: ARP MIRROR • Term: ARP BCAST on L3 Port • Term: ARP UCAST on L3 Port • Term: ARP UCAST on SVI
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) • Mirror: ISIS-L2 • Term: ISIS L2 - 1 • Term: ISIS L2 - 2 • Term: ISIS L2 - 3 • Term: ISIS L3 - 1 • Term: ISIS L3 - 2 • Term: ISIS L3 - 3
system-cpp-police-ospf-v4	• OSPFv2 MC • OSPFv2 UC

Class-map names	Associated feature(s)
system-cpp-police-ospf-v6	• OSPFv3 MC • OSPFv3 UC
system-cpp-police-bgp-v4	• BGP established IPv4 • BGP IPv4
system-cpp-police-bgp-v6	• BGP established IPv6 • BGP IPv6
system-cpp-police-ldp-v4	• LDP TCP DST IPv4 • LDP TCP SRC IPv4 • LDP UDP DST IPv4 • LDP UDP SRC IPv4
system-cpp-police-ldp-v6	• LDP TCP DST IPv6 • LDP TCP SRC IPv6 • LDP UDP DST IPv6 • LDP UDP SRC IPv6
system-cpp-police-rip-bc	• RIP • Term: RIP Broadcast
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	• GLBP IPv4 • HSRP IPv4 • VRRP IPv4
system-cpp-police-fhrp-v6	• GLBP IPv6 • HSRP IPv6 • VRRP IPv6

Class-map names	Associated feature(s)
system-cpp-police-bfd-v4	• BFD ECHO IPv4 • BFD IPv4 • BFD MHOP IPv4
system-cpp-police-bfd-v6	• BFD ECHO IPv6 • BFD IPv6 • BFD MHOP IPv6
system-cpp-police-mpls-oam	MPLS OAM IPv4
system-cpp-police-mcast-ctrl-v4	• MC NON ROUTABLE(v4) • MDT data UDP IPv4 • PIM IPv4
system-cpp-police-mcast-ctrl-v6	• AH IPV6 MC • ESP IPV6 MC • MC NON ROUTABLE(v6) • MDT data UDP IPv6 • PIM IPV6
system-cpp-police-l2-control	• CISCO Protocols • Fabric Wireless IEEE 802.3 XID • LLDP • MACSEC • Term: Cisco Protocols • Term: Fabric Wireless IEEE 802.3 XID • Term: L2PT • Term: LLDP • Term: MACSEC • Term: WMACSEC
system-cpp-police-mac-learning	Direct trap: MAC-LEARN / ETHERNET_LEARN_PUNT
system-cpp-police-topology-control	• STP (IEEE) • Term: IEEE STP

Class-map names	Associated feature(s)
system-cpp-police-protocol-snoop-v4	• ARP Snooping • IGMP I2 Ipts • IPv4 IGMP • Pim Hello Ipts • Term: IGMP SNOOP ENABLE - VLAN • Term: PIMv4 ENABLE - VLAN • Term: PIMv6 ENABLE - VLAN
system-cpp-police-protocol-snoop-v6	• IPv6 Options • SISF
system-cpp-police-dhcp-snooping	• DHCP Snooping • Egress DHCP Snooping
system-cpp-police-lacp	• LACP • Term: LACP
system-cpp-police-mtu-ttl-fail	• IPv6 Header Error • MPLS-VPN-TTL1 • MTU (L3) • TTL0 (MPLS) • TTL1 (L3) • TTL1 (MPLS)
system-cpp-police-meraki-nt-v4	• Meraki NT IPv4 first • Meraki NT IPv4 Second
system-cpp-police-meraki-nt-v6	• Meraki NT IPv6 first • Meraki NT IPv6 Second
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

Class-map names	Associated feature(s)
system-cpp-police-dhcp-v4	• DHCP Client (v4) • DHCP Server (v4) • DHCP(mirror) • Mirror: DHCP MIRROR • Term: DHCP IPv4 C to S - L3 • Term: DHCP IPv4 S to C - L3 • Term: DHCP IPv4 S to S - L3
system-cpp-police-dhcp-v6	• DHCP Client (v6) • DHCP Server (v6) • Term: DHCP IPv6 C to S - L3 • Term: DHCP IPv6 S to C - L3 • Term: DHCP IPv6 S to S - L3
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 • MC PIM RPF FAIL • MC SNOOP RPF FAIL • Mirror: MC SVI RPF FAIL
system-cpp-police-mcast-data	• MC DIRECT CONNECT • MC FWD DISABLED (v4) • MC FWD DISABLED (v6) • MC G PIM PUNT • MC NOT FOUND • MC SG PIM PUNT • MC SNOOP and DIRECT CONNECT • Mirror: MCAST *,G or S,G

Class-map names	Associated feature(s)
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 • GLEAN • IP Options(v4) Rx • IP Options(v4) Tx • IPv6 DEST-OPT • IPv6 HOP-OPT • NAT • 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	• TCP DST PORT LISP IPv4 • TCP SRC PORT LISP IPv4 • UDP DST PORT LISP IPv4 • UDP SRC PORT LISP IPv4
system-cpp-police-alt-sw-forward	• IPv4 MC • IPv6 MC • IPv6 UC
system-cpp-police-host-learning	Mirror: SDA-HOST-LEARN
system-cpp-police-svl-ipc	Direct trap: SVL_CONTROL_IPC
system-cpp-police-svl-ctrl	• Direct trap: L2_LPTS_TRAP7 • Direct trap: SVL_CONTROL_PROTOCOL • Term: SVL_DAD • Term: SVL_LACP

Class-map names	Associated feature(s)
system-cpp-police-mdns	• MDNS • Term: IPV4 MDNS • Term: IPV6 MDNS

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.

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 <code>cpp system-default</code> 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.

Software upgrades and CoPP

Software version upgrades and CoPP is a process that

- involves automatic system checks and updates for CoPP when upgrading software versions
- requires specific user actions before or after the upgrade activity depending on the upgrade method, and
- ensures that any configuration updates are reflected correctly and CoPP continues to work as expected.

System and user actions for upgrades

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.

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 – 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: <code>system-cpp-police-control-low-priority</code>).	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 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
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.

Restrictions for CoPP

Restrictions for control plane policing (CoPP) include:

- 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 system automatically reverts to the default policer rate after system bootup in order to protect the system bring-up process.
- User-defined CoPP class-maps and custom CoPP policies are not supported. Use **show running-config** to view current CoPP configuration, and use **show policy-map system-cpp-policy** or **show policy-map control-plane** to view default system CoPP policy details.

Configure CoPP

Enable a CPU queue and change the policer rate

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

Procedure

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-forus
```

```
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*.

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) Disable a CPU queue

Use this procedure to disable a CPU queue to prevent processing of specific traffic types in CoPP.

Procedure

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-forus
```

```
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 the configured policer for the class-map. This step is not recommended.

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.

Set 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

-
- 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

Example: Enable and change 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: Disable 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: Set the default policer rates for all CPU queues

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

```
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
```

Monitor 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 active punt entries	Displays the policies, the priorities, and the rates configured for the various traffic types. Note Mapping of punt entries may change across releases and platforms. We recommend that you run this command to verify the mapping.

Command	Purpose
show platform software fed active punt asic-cause brief	Displays FED punt-path statistics grouped by ASIC punt source and ASIC punt cause. The Source and Cause fields identify the hardware-origin punt classification for packets received by FED from the ASIC. The Drop counters, if present, indicate drops in FED punt processing for that source or cause. Note We recommend that you check the output of this command more than once to observe the exact change in statistics.
show platform software fed active punt ios-cause brief	Displays FED punt-path statistics grouped by IOS punt cause. The output shows how punted packets are classified after FED maps the hardware punt information into IOS software punt causes, along with received and dropped packet counters for each cause. Note We recommend that you check the output of this command more than once to observe the exact change in statistics.

This is a sample output of the **show policy-map control-plane** command:

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

Control Plane

Service-policy input: system-cpp-policy

Class-map: system-cpp-default (match-any)
  0 packets, 0 bytes
  5 minute offered rate 0000 bps, drop rate 0000 bps
  Match: none
  police:
    rate 2000 pps, burst 11264 packets
    conformed 8064 bytes; actions:
      transmit
    exceeded 0 bytes; actions:
      drop

Class-map: system-cpp-default-v4 (match-any)
  0 packets, 0 bytes
  5 minute offered rate 0000 bps, drop rate 0000 bps
  Match: none
  police:
    rate 2000 pps, burst 11264 packets
    conformed 0 bytes; actions:
      transmit
    exceeded 0 bytes; actions:
      drop

Class-map: system-cpp-default-v6 (match-any)
  0 packets, 0 bytes
  5 minute offered rate 0000 bps, drop rate 0000 bps
```

```

Match: none
police:
    rate 2000 pps, burst 11264 packets
    conformed 0 bytes; actions:
        transmit
    exceeded 0 bytes; actions:
        drop

Class-map: system-cpp-police-arp (match-any)
  0 packets, 0 bytes
  5 minute offered rate 0000 bps, drop rate 0000 bps
Match: none
police:
    rate 1000 pps, burst 11264 packets
    conformed 0 bytes; actions:
        transmit
    exceeded 0 bytes; actions:
        drop
.....
<output truncated>

```

This is a sample output of the **show policy-map system-cpp-policy** command:

Device# **show policy-map system-cpp-policy**

```

Policy Map system-cpp-policy
Class system-cpp-default
  police rate 2000 pps
    conform-action transmit
    exceed-action drop
Class system-cpp-default-v4
  police rate 2000 pps
    conform-action transmit
    exceed-action drop
Class system-cpp-default-v6
  police rate 2000 pps
    conform-action transmit
    exceed-action drop
Class system-cpp-police-arp
  police rate 1000 pps
    conform-action transmit
    exceed-action drop
Class system-cpp-police-icmp-v4
  police rate 2500 pps
    conform-action transmit
    exceed-action drop
Class system-cpp-police-icmp-v6
  police rate 2500 pps
    conform-action transmit
    exceed-action drop
Class system-cpp-police-isis
  police rate 1000 pps
    conform-action transmit
    exceed-action drop
Class system-cpp-police-ospf-v4
  police rate 2000 pps
    conform-action transmit
    exceed-action drop
Class system-cpp-police-ospf-v6
  police rate 2000 pps
    conform-action transmit
    exceed-action drop
.....
<output truncated>

```

This is a sample output of the **show platform software fed active punt entries** command:

Device# **show platform software fed active punt entries**

Punject Punt Entries

Source	Name	ASIC	Pri	TC	Policy
CIR-SW	CIR-HW	Rx Pkts		Drop Pkts	
		cur	delta	cur	delta
TRAP	ACL Drop(ETH)		0	1	0
	2000 2002	64	64	0	0
MIRROR	ARP		0	4	4
	1000 1001	0	0	0	0
TRAP	No Termination on L3 Port		0	80	0
	200 143	0	0	0	0
TRAP	CISCO Protocols		0	3	5
	16000 16021	63258	63258	0	0
TRAP	DHCP Client(v4)		0	3	4
	6000 6008	0	0	0	0
TRAP	DHCP Server(v4)		0	3	4
	6000 6008	0	0	0	0
TRAP	DHCP Client(v6)		0	3	4
	6000 6008	0	0	0	0
TRAP	DHCP Server(v6)		0	3	4
	6000 6008	0	0	0	0
TRAP	SmartPower/TCPMSS		0	4	4
	2000 2002	64	64	0	0
TRAP	ETH HOP-OPT		0	88	3
	2000 2002	0	0	0	0
MIRROR	ISIS (L2)		0	3	4
	1000 6008	0	0	0	0
TRAP	LLDP		0	4	5
	16000 16021	63258	63258	0	0
TRAP	STP(IEEE)		0	4	5
	5000 4863 546038 546038			0	0
TRAP	Fabric Wireless IEEE 802.3 XID		0	4	5
	16000 16021	63258	63258	0	0
TRAP	Pim Hello lpts		0	8	4
	2000 2002	0	0	0	0
TRAP	Multicast SSDP		0	8	4
	200 143	0	0	0	0
TRAP	Meraki LSP		0	3	5
	2000 2002	64	64	0	0
TRAP	IEEE (FC) DROP		0	0	0
	200 143	0	0	0	0
TRAP	GOLD		0	0	3
	2000 2002	64	64	0	0
TRAP	LACP		0	4	5
	600 572	0	0	0	0
TRAP	MACSEC		0	3	4
	16000 16021	63258	63258	0	0
TRAP	MC snoop lookup miss		0	2	0
	0 0	0	0	0	0
TRAP	GACL webauth punt (http/https		0	3	4
	2000 2002	64	64	0	0
TRAP	Egres DHCP Snooping		0	3	3
	1000 1001	0	0	0	0
TRAP	ARP Ucast on SVI		0	4	4
	1000 1001	0	0	0	0
TRAP	FQDN Redirect ACL punt (DNS re		0	3	4
	2000 2002	64	64	0	0

.....
<output truncated>

This is a sample output of the **show platform software fed active punt asic-cause brief** command:

Device# **show platform software fed active punt asic-cause brief**

ASIC Cause Statistics Brief

Source		Cause		Rx		Drop	
				cur	delta	cur	delta
UKNWN	DEFAULT			500	500	500	500
LPTS	ICMP IPv6 MC			21	21	0	0
LPTS	IPv6 Options			24	24	18	18
ITRAP	CISCO Protocols			63215	63215	0	0
ITRAP	STP(IEEE)			545620	545620	0	0
ITRAP	GOLD			64	64	0	0

This is a sample output of the **show platform software fed active punt ios-cause brief** command:

Device# **show platform software fed active punt ios-cause brief**

Statistics for all causes

Cause	Cause Info	Rcvd	Dropped
0	Reserved	0	518
7	ARP request or response	21	0
58	Layer2 bridge domain data pack	6	0
96	Layer2 control protocols	608849	0