CHAPTER 51

Configuring Control Plane Policing and Layer 2 Control Packet QoS

Note CoPP is supported on the following: Supervisor 6-E and Catalyst 4900M starting with Cisco IOS Release 12.2(50)SG; Supervisor 6L-E starting with Cisco IOS Release 12.2(52)X0; Catalyst 4948-E starting with Cisco IOS Release 12.2(54)X0; Supervisor Engine 7-E starting with Cisco IOS XE 3.1.0SG; Supervisor Engine 7L-E starting with Cisco IOS XE 3.2.0XO; Supervisor Engine 8-E starting with Cisco IOS XE 3.3.0SG.

This chapter contains information on how to protect your Catalyst 4500 series switch using control plane policing (CoPP). The information covered in this chapter is unique to the Catalyst 4500 series switches, and it supplements the network security information and procedures in Chapter 54, "Configuring Network Security with ACLs." This information also supplements the network security information and procedures in these publications:

- *Cisco IOS Security Configuration Guide*, *Cisco IOS Release 12.4*, at this URL:

  http://www.cisco.com/en/US/docs/ios/security/configuration/guide/12_4/sec_12_4_book.html
- *Cisco IOS Security Command Reference*, *Cisco IOS Release 12.4*, at this URL:

  http://www.cisco.com/en/US/docs/ios/security/command/reference/sec_book.html

This chapter includes the following major sections:

Note For complete syntax and usage information for the switch commands used in this chapter, see the *Cisco Catalyst 4500 Series Switch Command Reference* and related publications at this location:

http://www.cisco.com/en/US/products/hw/switches/ps4324/index.html

If a command is not in the *Catalyst 4500 Series Switch Command Reference*, you can locate it in the Cisco IOS library. See the *Cisco IOS Command Reference* and related publications at this location:

http://www.cisco.com/en/US/products/ps6350/index.html

Configuring Control Plane Policing

This section includes these topics:

About Control Plane Policing

Note Catalyst 4500 switch support hardware CoPP for all IPv6 First Hop Security Features (DHCPv6 Inspection/Guard, DHCPv6 remote-ID option for Layer 2, IPv6 full RA Guard, ...) However, due to inability of VFE to match ICMP v6 packets for policing in the outward direction, hardware CoPP does not work on Supervisor 6-E, Supervisor 6L-E, Catalyst 4900M, and Catalyst 4948-E

The control plane policing (CoPP) feature increases security on the Catalyst 4500 series switch by protecting the CPU from unnecessary or DoS traffic and giving priority to important control plane and management traffic. The classification TCAM and QoS policers provide CoPP hardware support.

Traffic managed by the CPU is divided into three functional components or *planes*:

- Data plane
- Management plane
- Control plane

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

By default, you receive a list of predefined ACLs matching a selected set of Layer 2 and Layer 3 control plane packets. You can further define your preferred policing parameters for each of these packets and modify the matching criteria of these ACLs.

The following table lists the predefined ACLs.

Predefined Named ACL	Description
system-cpp-dot1x	MAC DA = 0180.C200.0003
system-cpp-lldp	MAC DA = 0180.C200.000E
system-cpp-mcast-cfm	MAC DA = 0100.0CCC.CCC0 - 0100.0CCC.CCC7
system-cpp-ucast-cfm	MAC DA = 0100.0CCC.CCC0
system-cpp-bpdu-range	MAC DA = 0180.C200.0000 - 0180.C200.000F
system-cpp-cdp	MAC DA = 0100.0CCC.CCCC (UDLD/DTP/VTP/Pagp)
system-cpp-sstp	MAC DA = 0100.0CCC.CCCD

Predefined Named ACL	Description
system-cpp-cgmp	MAC DA = 01.00.0C.DD.DD.DD
system-cpp-hsrpv2	IP Protocol = UDP, IPDA = 224.0.0.102
system-cpp-ospf	IP Protocol = OSPF, IP DA matches 224.0.0.0/24
system-cpp-igmp	IP Protocol = IGMP, IP DA matches 224.0.0.0/3
system-cpp-pim	IP Protocol = PIM, IP DA matches 224.0.0.0/24
system-cpp-all-systems-on-subnet	IP DA = 224.0.0.1
system-cpp-all-routers-on-subnet	IP DA = 224.0.0.2
system-cpp-ripv2	IP DA = 224.0.0.9
system-cpp-ip-mcast-linklocal	IP DA = 224.0.0.0/24
system-cpp-dhcp-cs	IP Protocol = UDP, L4SrcPort = 68, L4DstPort = 67
system-cpp-dhcp-sc	IP Protocol = UDP, L4SrcPort = 67, L4DstPort = 68
system-cpp-dhcp-ss	IP Protocol = UDP, L4SrcPort = 67, L4DstPort = 67

For the data and management plane traffic, you can define your own ACLs to match the traffic class that you want to police.

CoPP uses MQC to define traffic classification criteria and to specify the configurable policy actions for the classified traffic. MQC uses class maps to define packets for a particular traffic class. After you have classified the traffic, you can create policy maps to enforce policy actions for the identified traffic. The **control-plane global** configuration command allows you to directly attach a CoPP service policy to the control plane.

The policy map system-cpp-policy must contain the predefined class maps in the predefined order at the beginning of the policy map. The best way to create system-cpp-policy policy map is by using the global macro system-cpp.

The system-cpp-policy policy map contains the predefined class maps for the control plane traffic. The names of all system-defined CoPP class maps and their matching ACLs contain the prefix system-cpp-. By default, no action is specified for each traffic class. You can define your own class maps matching CPU-bound data plane and management plane traffic. You can also add your defined class maps to system-cpp-policy.

General Guidelines for Control Plane Policing

Guidelines for control plane policing include the following:

- Port security might cancel the effect of CoPP for non-IP control packets.

  Although source MAC learning on a Catalyst 4500 series switch is performed in software, learning control packets' source MAC addresses (for example, IEEE BPDU, CDP, SSTP BPDU, GARP/) is not allowed. Once you configure port security on a port where you expect a high rate of potentially unanticipated control packets, the system generates a copy of the packet to the CPU (until the source address is learned), instead of forward it.

  The current architecture of the Catalyst 4500 supervisor engine does not allow you to apply policing on the copy of packets sent to the CPU. You can only apply policing on packets that are forwarded to the CPU. Copies of packets are sent to the CPU at the same rate as control packets, and port security is not triggered because learning from control packets is not allowed. Policing is not applied because the packet copy, not the original, is sent to the CPU.

- ARP policing is not supported on either the classic series supervisor engines or fixed configuration switches. It is supported on the Catalyst 4900M and 4948E switches, Supervisor Engine 6-E, and Supervisor Engine 6L-E (use “match protocol arp” to classify).
- Only ingress CoPP is supported. So only input keyword is supported in control-plane related CLIs.
- Use ACLs and class-maps to identify data plane and management plane traffic that are handled by CPU.
- The only action supported in CoPP policy-map is police.
- Do not use the log keyword in the CoPP policy ACLs.

Default Configuration

CoPP is disabled by default.

Configuring CoPP for Control Plane Traffic

To configure CoPP for control plane traffic, perform this task:

	Command	Purpose
Step 1	`Switch# config terminal`	Enters global configuration mode.
Step 2	`Switch(config)# qos`	(Optional) Enables QoS globally.
Step 3	`Switch(config)#` `macro global apply system-cpp`	(Optional) Creates the system-cpp-policy policy map and attaches it to the control plane.
Step 4	`Switch(config)# policy-map system-cpp-policy` `Switch(config-pmap)# class {system-cpp-dot1x | system-cpp-bpdu-range | system-cpp-cdp | service | system-cpp-sstp | system-cpp-cgmp | system-cpp-ospf | system-cpp-igmp | system-cpp-pim | system-cpp-all-systems-on-subnet | system-cpp-all-routers-on-subnet | system-cpp-ripv2 | system-cpp-hsrpv2 | system-cpp-ip-mcast-linklocal | system-cpp-dhcp-cs | system-cpp-dhcp-sc | system-cpp-dhcp-ss}` `Switch(config-pmap-c)# police [aggregate name] rate burst [conform-action {drop | transmit}] [{exceed-action {drop | transmit}}]}`	Associates actions to one or multiple system-defined control plane traffic in the service policy map. Repeat this step if necessary.
Step 5	`Switch# show policy-map system-cpp-policy`	(Optional) Verifies the configuration.

The following example shows how to police CDP packets:

```
Switch# config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# qos
Switch(config)# macro global apply system-cpp
Switch(config)# policy-map system-cpp-policy
Switch(config-pmap)# class system-cpp-cdp
Switch(config-pmap-c)# police 32000 1000 conform-action transmit exceed-action drop
Switch(config-pmap-c)# end
```

```
Switch# show policy-map system-cpp-policy
  Policy Map system-cpp-policy
    Class system-cpp-dot1x
    Class system-cpp-bpdu-range
*    Class system-cpp-cdp
      police 32000 bps 1000 byte conform-action transmit exceed-action drop
    Class system-cpp-sstp
    Class system-cpp-cgmp
    Class system-cpp-ospf
    Class system-cpp-hsrpv2
    Class system-cpp-igmp
    Class system-cpp-pim
    Class system-cpp-all-systems-on-subnet
    Class system-cpp-all-routers-on-subnet
    Class system-cpp-ripv2
    Class system-cpp-ip-mcast-linklocal
    Class system-cpp-dhcp-cs
    Class system-cpp-dhcp-sc
    Class system-cpp-dhcp-ss
Switch#
```

Configuring CoPP for Data Plane and Management Plane Traffic

To configure CoPP for data plane and management plane traffic, perform this task:

	Command	Purpose
Step 1	`Switch(config)# qos`	(Optional) Enables QoS globally.
Step 2	`Switch(config)#` `macro global apply system-cpp`	(Optional) Attaches the system-cpp-policy policy map to the control plane.

	Command	Purpose

Step 3	`Switch(config)# {ip | mac} access-list extended {access-list-name}`
	
	`For an ip access list, issue`
	`Switch(config-ext-nacl)#{permit|deny} {protocol} source {source-wildcard} destination {destination-wildcard}`
	
	`For a mac access list, issue`
	`Switch(config-ext-macl)#{permit|deny} source {source-wildcard} destination {destination-wildcard} [protocol-family]`
	
	`OR`
	
	`Switch(config)# access-list {access-list-name} {permit | deny} {type-code wild-mask | address mask}`
		Defines ACLs to match traffic:
		• **permit—**Sets the conditions under which a packet passes a named ACL
		• **deny**—Sets the conditions under which a packet does not pass a name ACL
		Note You must configure ACLs in most cases to identify the important or unimportant traffic.
		• **type-code—**16-bit hexadecimal number written with a leading 0x; for example, 0x6000. Specify either a Link Service Access Point (LSAP) type code for 802-encapsulated packets or a SNAP type code for SNAP-encapsulated packets. (LSAP, sometimes called SAP, refers to the type codes found in the DSAP and SSAP fields of the 802 header.)
		• **wild-mask**—16-bit hexadecimal number whose ones bits correspond to bits in the type-code argument. The wild-mask indicates which bits in the type-code argument should be ignored when making a comparison. (A mask for a DSAP/SSAP pair should always be 0x0101 because these two bits are used for purposes other than identifying the SAP code.)
		• **address**—48-bit Token Ring address written as a dotted triple of four-digit hexadecimal numbers. This field is used for filtering by vendor code.
		• **mask—**48-bit Token Ring address written as a dotted triple of four-digit hexadecimal numbers. The ones bits in the mask are the bits to be ignored in address. This field is used for filtering by vendor code.

Step 4	`Switch(config)# class-map {traffic-class-name}`
	
	`Switch(config-cmap)# match access-group {access-list-number | name {access-list-name}}`
		Defines the packet classification criteria. To identify the traffic associated with the class, use the **match** statements.

Step 5	`Switch(config-cmap)# exit`	Returns to global configuration mode.

	Command	Purpose

Step 6	`Switch(config)# policy-map system-cpp-policy`
	
	`Switch(config-pmap)# class {class-map-name}`
	
	`Switch(config-pmap-c)# police [aggregate name] rate burst [conform-action {drop | transmit}] [{exceed-action {drop | transmit}}]`
		Adds the traffic classes to the CoPP policy map. Uses the **police** statement to associate actions to the traffic class.

Step 7	`Switch(config)# end`	Returns to privileged EXEC mode.
Step 8	`Switch# show policy-map system-cpp-policy`	Verifies your entries.

The following example shows how to configure trusted hosts with source addresses 10.1.1.1 and 10.1.1.2 to forward Telnet packets to the control plane without constraint, while allowing all remaining Telnet packets to be policed at the specific rate. This example assumes that global QoS is enabled and that the system-cpp-policy policy map was created.

```
Switch# config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# qos
Switch(config)# macro global apply system-cpp

! Allow 10.1.1.1 trusted host traffic.
Switch(config)# access-list 140 deny tcp host 10.1.1.1 any eq telnet

! Allow 10.1.1.2 trusted host traffic.
Switch(config)# access-list 140 deny tcp host 10.1.1.2 any eq telnet

! Rate limit all other Telnet traffic.
Switch(config)# access-list 140 permit tcp any any eq telnet

! Define class-map "telnet-class."
Switch(config)# class-map telnet-class
Switch(config-cmap)# match access-group 140
Switch(config-cmap)# exit

! Add the class-map "telnet-class" to "system-cpp-policy" and define the proper action
Switch(config)# policy-map system-cpp-policy
Switch(config-pmap)# class telnet-class
Switch(config-pmap-c)# police 80000 1000 conform transmit exceed drop
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit

! Verify the above configuration steps
Switch# show policy-map system-cpp-policy
  Policy Map system-cpp-policy
    Class system-cpp-dot1x
    Class system-cpp-bpdu-range
    Class system-cpp-cdp
      police 32000 bps 1000 byte conform-action transmit exceed-action drop
    Class system-cpp-sstp
    Class system-cpp-cgmp
    Class system-cpp-ospf
    Class system-cpp-hsrpv2
    Class system-cpp-igmp
    Class system-cpp-pim
    Class system-cpp-all-systems-on-subnet
```

```
    Class system-cpp-all-routers-on-subnet
    Class system-cpp-ripv2
    Class system-cpp-ip-mcast-linklocal
    Class system-cpp-dhcp-cs
    Class system-cpp-dhcp-sc
    Class system-cpp-dhcp-ss
*    Class telnet-class
      police 80000 1000 byte conform-action drop exceed-action drop
```

Control Plane Policing Configuration Guidelines and Restrictions

When using (or configuring) control plane policing, consider these guidelines and restrictions:

All supervisor engines

When configuring CoPP, consider these guidelines:

- Only ingress CoPP is supported. Only the **input** keyword is supported in control plane-related CLIs.
- Control plane traffic can be policed only through CoPP. Traffic cannot be policed at the input interface or VLAN even though a policy map containing the control plane traffic is accepted when the policy map is attached to an interface or VLAN.
- Use ACLs and class maps to identify data plane and management plane traffic that are handled by the CPU. U[1]ser defined class maps should be added to the system-cpp-policy policy map for CoPP.
- The default system-cpp-policy policy map does not define actions for the system-defined class maps (no policing).
- The only action supported in system-cpp-policy is police.
- You can use both MAC and IP ACLs to define data plane and management plane traffic classes. However, if a packet also matches a predefined ACL for the control plane traffic, a police (or no police) action will operate on the control plane class because the control plane classes appear above the user-defined classes in the service policy.
- The exceeding action **policed-dscp-transmit** is not supported for CoPP.
- Do not use the **log** keyword in CoPP policy ACLs. Instead, if you want to determine if rogue packets are arriving, view the output of the **show policy-map interface** command or use the span feature.

Do not apply to Catalyst 4900M, Catalyst 4948E, Supervisor Engine 6-E, and Supervisor Engine 6L-E

- To police control plane traffic, use the system-defined class maps.
- System-defined class maps cannot be used in policy maps for regular QoS.
- The policy map named system-cpp-policy is dedicated for CoPP.
- CoPP is not enabled unless global QoS is enabled and a police action is specified.

1.

Monitoring CoPP

You can enter the **show policy-map control-plane** command to develop site-specific policies, to monitor statistics for the control plane policy, and to troubleshoot CoPP. This command displays dynamic information about the actual policy applied, including rate information and the number of bytes (and packets) that conformed or exceeded the configured policies both in hardware and in software.

The output of the **show policy-map control-plane** command is similar to the following:

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

Control Plane

Service-policy input: system-cpp-policy

    Class-map: system-cpp-dot1x (match-all)
      0 packets
      Match: access-group name system-cpp-dot1x

    Class-map: system-cpp-bpdu-range (match-all)
      0 packets
      Match: access-group name system-cpp-bpdu-range

*    Class-map: system-cpp-cdp (match-all)
      160 packets
      Match: access-group name system-cpp-cdp
**      police: Per-interface
        Conform: 22960 bytes Exceed: 0 bytes
*
    Class-map: system-cpp-sstp (match-all)
      0 packets
      Match: access-group name system-cpp-sstp

    Class-map: system-cpp-cgmp (match-all)
      0 packets
      Match: access-group name system-cpp-cgmp

    Class-map: system-cpp-hsrpv2 (match-all)
      0 packets
      Match: access-group name system-cpp-hsrpv2

    Class-map: system-cpp-ospf (match-all)
      0 packets
      Match: access-group name system-cpp-ospf

    Class-map: system-cpp-igmp (match-all)
      0 packets
      Match: access-group name system-cpp-igmp

    Class-map: system-cpp-pim (match-all)
      0 packets
      Match: access-group name system-cpp-pim

    Class-map: system-cpp-all-systems-on-subnet (match-all)
      0 packets
      Match: access-group name system-cpp-all-systems-on-subnet

    Class-map: system-cpp-all-routers-on-subnet (match-all)
      0 packets
      Match: access-group name system-cpp-all-routers-on-subnet

    Class-map: system-cpp-ripv2 (match-all)
      0 packets
```

```
      Match: access-group name system-cpp-ripv2

    Class-map: system-cpp-ip-mcast-linklocal (match-all)
      0 packets
      Match: access-group name system-cpp-ip-mcast-linklocal

    Class-map: system-cpp-dhcp-cs (match-all)
      83 packets
      Match: access-group name system-cpp-dhcp-cs

    Class-map: system-cpp-dhcp-sc (match-all)
      0 packets
      Match: access-group name system-cpp-dhcp-sc

    Class-map: system-cpp-dhcp-ss (match-all)
      0 packets
      Match: access-group name system-cpp-dhcp-ss

Class-map: telnet-class (match-all)
      92 packets
      Match: access-group 140
      police:
          cir 32000 bps, bc 1500 bytes
        conformed 5932 bytes; actions:
          transmit
        exceeded 0 bytes; actions:
          drop
        conformed 0000 bps, exceed 0000 bps

Class-map: class-default (match-any)
      0 packets
      Match: any
        0 packets
Switch#
```

To clear the counters on the control plane, enter the **clear control-plane *** command:

```
Switch# clear control-plane *
Switch#
```

To display all the CoPP access list information, enter the **show access-lists** command:

```
Switch# show access-lists
Extended IP access list system-cpp-all-routers-on-subnet
10 permit ip any host 224.0.0.2
Extended IP access list system-cpp-all-systems-on-subnet
10 permit ip any host 224.0.0.1
Extended IP access list system-cpp-dhcp-cs
10 permit udp any eq bootpc any eq bootps Extended IP access list
system-cpp-dhcp-sc
10 permit udp any eq bootps any eq bootpc Extended IP access list
system-cpp-dhcp-ss
10 permit udp any eq bootps any eq bootps Extended IP access list
system-cpp-igmp
10 permit igmp any 224.0.0.0 31.255.255.255 Extended IP access list
system-cpp-ip-mcast-linklocal
10 permit ip any 224.0.0.0 0.0.0.255 Extended IP access list
system-cpp-ospf
10 permit ospf any 224.0.0.0 0.0.0.255 Extended IP access list
system-cpp-pim
10 permit pim any 224.0.0.0 0.0.0.255 Extended IP access list
system-cpp-ripv2
10 permit ip any host 224.0.0.9
Extended MAC access list system-cpp-bpdu-range
```

```
permit any 0180.c200.0000 0000.0000.000f Extended MAC access list
system-cpp-cdp
permit any host 0100.0ccc.cccc
Extended MAC access list system-cpp-cgmp
permit any host 0100.0cdd.dddd
Extended MAC access list system-cpp-dot1x
permit any host 0180.c200.0003
system-cpp-sstp
permit any host 0100.0ccc.cccd
```

To display one CoPP access list, enter the **show access-lists system-cpp-cdp** command:

```
Switch# show access-list system-cpp-cdp
Extended MAC access list system-cpp-cdp
permit any host 0100.0ccc.cccc
Switch#
```

Configuring Layer 2 Control Packet QoS

Layer 2 control packet QoS enables you to police control packets arriving on a physical port or LAN.

This section includes these topics:

Understanding Layer 2 Control Packet QoS

You might want to police incoming Layer 2 control packets such as STP, CDP, VTP, SSTP, BPDU, EAPOL and LLDP on a specific port before the packets reach CPU. This could serve as a first line of defense before aggregate traffic is subjected to policing (through CoPP). By default, policers cannot be applied to Layer 2 control packets in the input direction. This prevents users from inadvertently policing or dropping critical Layer 2 control packets.

While this approach protects a user who is wrongly policing control packets, it introduces a more serious problem. If a flood of Layer 2 control packets is received on any of the switch interfaces at a very high rate due to a DoS attack or to a loop introduced in the customer network because of misconfiguration, CPU utilization can increase quickly. This can have adverse impacts such as loss of protocol keep-alives and routing protocol updates. The Layer 2 control packet QoS feature allows you to police Layer 2 control packets at the port, VLAN, or port- VLAN level in the input direction.

Default Configuration

Layer 2 control packet QoS is disabled by default.

Enabling Layer 2 Control Packet QoS

To enable Layer 2 control packet QoS, perform this task:

	Command	Purpose
Step 1	`Switch# config terminal`	Enters configuration mode.
Step 2	`Switch(config)# [no] qos control-packets [bpdu-range | cdp-vtp | eapol | sstp | protocol-tunnel | lldp]`	Enables QoS on all or a specific packet type. Use the **no** keyword to disable QoS on all or a specific packet type.
Step 3	`Switch(config)# end`	Exits configuration mode.
Step 4	`Switch# show run | inc qos control-packets`	Verifies the configuration.

Table 51-1 lists the types of packets impacted by this feature.

Table 51-1 Packet Type and Actionable Address Range

Type of Packet that the Feature is Enabled On	Range of Address the Feature Acts On
BPDU-range	0180.C200.0000 BPDU 0180.C200.0002 OAM, LACP 0180.C200.0003 EAPOL
CDP-VTP	0100.0CCC.CCCC
SSTP	0100.0CCC.CCCD
LLDP	0180.C200.000E

The following example shows how to enable QoS for CDP packets and to apply a policer to CDP packets arriving on interface gi3/1 and VLAN 1:

```
Switch# config terminal
Switch(config)# qos control-packets cdp-vtp
Switch(config)# end
Switch# show run | inc qos control-packets
qos control-packets cdp-vtp
Switch# show class-map
Class Map match-any system-control-packet-cdp-vtp (id 1)

       Match access-group name system-control-packet-cdp-vtp

Create a policy map and attach it to interface gi3/1 , vlan 1
Switch# config terminal
Switch(config)# policy-map police_cdp
Switch(config-pmap)# class system-control-packet-cdp-vtp
Switch(config-pmap-c)# police 32k
Switch(config-pmap-c)# end

Switch(config)# interface gi3/1
Switch(config-if)# vlan 1
Switch(config-if-vlan-range)# service-policy in police_cdp
```

```
Switch(config-if-vlan-range)# exit
Switch(config-if)# exit
Switch(config)# exit
Switch# show policy-map interface gi3/1

GigabitEthernet3/1 vlan 1

  Service-policy input: police_cdp

    Class-map: system-control-packet-cdp-vtp (match-any)
      0 packets
      Match: access-group name system-control-packet-cdp-vtp
        0 packets
      police:
          cir 32000 bps, bc 1500 bytes
        conformed 0 packets, 0 bytes; actions:
          transmit
        exceeded 0 packets, 0 bytes; actions:
          drop
        conformed 0000 bps, exceed 0000 bps

    Class-map: class-default (match-any)
      0 packets
```

Disabling Layer 2 Control Packet QoS

The **no qos control-packet** command disables QoS for all packet types.

The following example shows how to disable QoS for CDP packets after QoS is enabled for all packet types:

```
Switch# show running-configuration | include qos control-packets
qos control-packets bpdu-range
qos control-packets cdp-vtp
qos control-packets lldp
qos control-packets eapol
qos control-packets sstp
qos control-packets protocol-tunnel
```

Note When all control packets (CDP/VTP, bpdu-range, SSTP, LLDP, and protocol-tunnel), are enabled only qos control-packets is nevgen'd. Individual protocol names mentioned in the previous output are nvegen'd only if the some of the control packets are configured.

```
Switch# config terminal
Switch(config)# no qos control-packets cdp-vtp
Switch(config)# end
Switch# show running-configuration | include qos control-packets
qos control-packets bpdu-range
qos control-packets lldp
qos control-packets sstp
qos control-packets protocol-tunnel
```

Note When you unconfigure this feature for a specified protocol type, the user-configured policies handling that protocol type immediately become ineffective. To save TCAM resources, remove the policies as well as MACLs and class maps (auto-generated or user-defined).

Note TCAM resources are not consumed when the interface is in a down state.

Table 51-2 displays the auto-generated MACLs and class maps that are created when you enable the feature on the corresponding packet type.

Table 51-2 Packet Types and Auto-Generated MACL/Class Maps

Packet Type	Auto-Generated MACL/Class Map
BPDU-range	mac access-list extended system-control-packet-bpdu-range permit any 0180.c200.0000 0000.0000.000c class-map match-any system-control-packet-bpdu-range match access-group name system-control-packet-bpdu-range
SSTP	mac access-list extended system-control-packet-sstp permit any host 0100.0ccc.cccd class-map match-any system-control-packet-sstp match access-group name system-control-packet-sstp
CDP-VTP	mac access-list extended system-control-packet-cdp-vtp permit any host 0100.0ccc.cccc class-map match-any system-control-packet-cdp-vtp match access-group name system-control-packet-cdp-vtp
EAPOL	mac access-list extended system-control-packet-eapol permit any any 0x888E class-map match-any system-control-packet-eapol match access-group name system-control-packet-eapol
LLDP	mac access-list extended system-control-packet-lldp permit any host 0180.c200.000e class-map match-any system-control-packet-lldp match access-group name system-control-packet-lldp
PROTOCOL TUNNEL	mac access-list extended system-control-packet-protocol-tunnel permit any host 0100.0ccd.cdd0 class-map match-any system-control-packet-protocol-tunnel match access-group name system-control-packet-protocol-tunnel

Layer 2 Control Packet QoS Configuration Examples

You can use CoPP and Layer 2 control packet QoS together to prevent DoS attacks to the CPU. In the following example, BPDUs arriving on interface gi3/1, VLAN 1 and VLAN 2 are limited to 32 Kbps and 34 Kbps, respectively. Aggregate BPDU traffic to CPU then is further rate-limited to 50 Kbps using CoPP.

```
Switch(config)# qos control-packets
```

```
Switch(config)# policy-map police_bpdu_1
Switch(config-pmap)# class system-control-packet-bpdu-range
Switch(config-pmap-c)# police 32k 1000
Switch(config-pmap-c-police)# exit
Switch(config-pmap-c)# exit
Switch(config-pmap)# policy-map police_bpdu_2
Switch(config-pmap)# class system-control-packet-bpdu-range
Switch(config-pmap-c)# police 34k
Switch(config-pmap-c-police)# exit
```

Configuring Layer 2 Control Packet QoS

```
Switch(config)# interface gi3/1
Switch(config-if)# vlan-range 1
Switch(config-if-vlan-range)# service-policy in police_bpdu_1
Switch(config-if-vlan-range)# exit
Switch(config-if)# interface gi3/2
Switch(config-if)# vlan-range 2
Switch(config-if-vlan-range)# service-policy in police_bpdu_1
Switch(config-if-vlan-range)# exit
```

Configuring Control Plane Policy

```
Switch(config)# macro global apply system-cpp
Switch(config)# policy-map system-cpp-policy
Switch(config-pmap)# class system-cpp-bpdu-range
Switch(config-pmap-c)# police 50k
Switch(config-pmap-c-police)# exit
Switch(config-pmap-c)# exit
```

Note To reduce the consumption of policer resources, you can also use named-aggregate policers applied to a group of ports or VLANs.

Note Do not modify class maps and MACLs that are auto-generated by the system. This action can cause unexpected behavior when the switch reloads or when the running configuration is updated from a file.

To refine or modify system-generated class maps or MACLs, apply user-defined class maps and MACLs.

Note User defined class map names must begin with the prefix system-control-packet. If not, certain hardware (Catalyst 4924, Catalyst 4948, Catalyst 4948-10GE, Supervisor Engine II-Plus, Supervisor Engine II+10GE, Supervisor Engine V, and Supervisor Engine V-10GE) might not perform the configured QoS action.

For example, the following are valid user-defined class map names to police Layer 2 control packets because they begin with the prefix system-control-packet:

```
system-control-packet-bpdu1
system-control-packet-control-packet
```

No such restrictions exist on the names you can use for user-defined MACLs (access-groups).

The following example shows how to create user-defined MACLs and class maps to identify EAPOL and BPDU packets. Because the auto-generated class map system-control-packet-bpdu range matches three packet types (BPDU, EAPOL, and OAM), policing this traffic class affects all three packet types. To police BPDU and EAPOL packets at different rates, you can set user-defined MACL and class map as follows:

```
Switch(config)# mac access-list extended system-control-packet-bpdu
```

```
Switch(config-ext-macl)# permit any host 0180.c200.0000
Switch(config-ext-macl)# exit
Switch(config)# class-map match-any system-control-packet-bpdu
Switch(config-cmap)# match access-group name system-control-packet-bpdu
Switch(config-cmap)# exit

Switch(config)# mac access-list extended system-control-packet-eapol
Switch(config-ext-macl)# permit any host 0180.c200.0003
Switch(config-ext-macl)# exit
Switch(config)# class-map match-any system-control-packet-eapol
Switch(config-cmap)# match access-group name system-control-packet-eapol
Switch(config-cmap)# exit
```

Layer 2 Control Packet QoS Guidelines and Restrictions

When using (or configuring) Layer 2 control packet QoS, consider these guidelines and restrictions:

- When you enable Layer 2 control packet QoS, it applies to all ports on the switch. If Layer 2 control packets are not explicitly classified in the policy attached to port or VLAN, the actions in class-default will be applied as per normal QoS rules.
- Place classifiers that match control packets at the beginning of a policy map followed by other traffic classes, ensuring that Layer 2 control packets are not subjected to inadvertent QoS actions.
- The application of default class (class-default) actions depends on the type of supervisor engine:
  - Supervisor Engine V-10GE with NetFlow support—Actions associated with class-default are never applied on unmatched control packets; a default permit action is applied. Only actions associated with class maps that begin with system-control-packet are applied on control packets.
  - All other supervisor engines—Actions associated with class-default are applied on unmatched control packets.
- If you enable the feature on a BPDU range, EAPOL packets are policed only after the initial 802.1X authentication phase completes.

Policing IPv6 Control Traffic

On Catalyst 4900M, Catalyst 4948E, Supervisor Engine 6-E, and Supervisor Engine 6L-E, IPv6 control packets such as OSPF, PIM and MLD can be policed on a physical port, VLAN, or control plane by configuring IPv6 ACLs to classify such traffic and then applying a QoS policy to police such traffic.

The following examples show how to police OSPFv6, PIMv6 and MLD control traffic received on a port.

This example shows how to configure a traffic class to identify OSPFv6 control packets by its destination IP v6 address:

```
Switch# config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# ipv6 access-list ospfv6
Switch(config-ipv6-acl)# permit ipv6 any host ff02:5
Switch(config-ipv6-acl)# exit
Switch(config)# class-map ospfv6Class
Switch(config-camp)# match access-group name ospfv6
Switch(config-camp)# exit
```

The following example shows how to configure a traffic class to identify PIMv6 control packets by its destination IPv6 address:

```
Switch(config)# ipv6 access-list pimv6
Switch(config-ipv6-acl)# permit ipv6 any host ff02::d
Switch(config-ipv6-acl)# exit
Switch(config)# class-map pimv6Class
Switch(config-cmap)# match access-group name pimv6
Switch(config-cmap)# exit
```

The following example shows how to configure a traffic class to identify MLD protocol control packets:

```
Switch(config)# ipv6 access-list mldv1
Switch(config-ipv6-acl)# permit icmp any any mld-query
Switch(config-ipv6-acl)# permit icmp any any mld-report
Switch(config-ipv6-acl)# permit icmp any any mld-reduction
Switch(config-ipv6-acl)# exit
Switch(config)# class-map mldClass
Switch(config-cmap)# match access-group name mldv1
Switch(config-cmap)# exit
```

The following example shows how to configure a QoS policy to police OSPFv6, PIMv6 and MLD traffic classes:

```
Switch(config)# policy-map v6_control_packet_policy
Switch(config-pmap)# class mldClass
Switch(config-pmap-c)# police 32k
Switch(config-pmap-c-police)# class ospfv6Class
Switch(config-pmap-c)# police 32k
Switch(config-pmap-c)# class pimv6Class
Switch(config-pmap-c)# police 32k
Switch(config-pmap-c-police)# exit
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# exit
Switch# show policy-map

Policy Map v6_control_packet_policy
    Class mldClass
     police cir 32000 bc 1500
       conform-action transmit
       exceed-action drop
    Class ospfv6Class
     police cir 32000 bc 1500
       conform-action transmit
       exceed-action drop
    Class pimv6class
     police cir 32000 bc 1500
       conform-action transmit
       exceed-action drop
```

The following example shows how to policy to interface gi2/2 in the input direction:

```
Switch# config terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface gi2/2
Switch(config-if)# service-policy in v6_control_packet_policy
Switch(config-if)# exit
```