QoS Configuration Guide, Cisco Catalyst IE3x00 Rugged, IE3400 Heavy Duty, and ESS3300 Series Switches

Cisco Catalyst IE3200 Rugged Series
Updated March 27, 2026

© 2020-2026 Cisco Systems, Inc. All rights reserved.

Full Cisco Trademarks with Software License

Full Cisco Trademarks with Software License

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: https://www.cisco.com/c/en/us/about/legal/trademarks.html. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

1 Configure QoS

Topics:

- QoS configuration

Configure Quality of Service (QoS) settings to optimize network traffic and prioritize critical applications.

QoS configuration

Describes the process of managing network traffic priority using modular QoS command-line interface commands. This configuration ensures specific traffic streams receive appropriate bandwidth and latency treatment on Cisco IE-3x00 and ESS3300 switch platforms.

QoS configuration provides a hierarchical framework for prioritizing or limiting specific traffic streams to ensure network performance.

The QoS configuration process includes the following characteristics:

- Provides preferential treatment to certain types of traffic at the expense of others.
- Uses modular QoS command-line interface (MQC) commands for implementation.
- Provides a comprehensive hierarchical configuration framework for prioritizing or limiting specific streams of traffic.

QoS behavior without configuration

When QoS is not configured, the switch offers best-effort service to each packet, regardless of the packet contents or size. It sends the packets without any assurance of reliability, delay bounds, or throughput.

Modular QoS CLI

Describes a command-line interface that allows users to create traffic policies and attach these policies to interfaces for quality of service management.

Modular QoS CLI (MQC) is a command-line interface that

- allows you to create traffic policies and attach them to interfaces,
- uses traffic classes to classify traffic, and
- applies QoS features in traffic policies to determine how to treat the classified traffic.

Modular QoS CLI configuration steps

Modular QoS CLI configuration includes these steps:

1. **Define a traffic class.**

   Use the **class-map [match-all | match-any]** *class-map-name* global configuration command to define a traffic class and to enter class-map configuration mode. A traffic class contains three elements: a name; an instruction on how to evaluate the configured match commands (if more than one match command is configured in the class map); and a series of match commands.

   - You name the traffic class in the class-map command line to enter class-map configuration mode.
   - You can optionally include keywords to evaluate these match commands by entering class-map match-any or class-map match-all. If you specify match-any, the traffic being evaluated must match one of the specified criteria. If you specify match-all, the traffic being evaluated must match all of the specified criteria. A match-all class map can contain only one match statement. A match-any class map can contain multiple match statements.

   If you do not enter match-all or match-any, the default is to match all.

   - You use the match class-map configuration commands to specify criteria for classifying packets. If a packet matches the specified criteria, that packet is considered a member of the class and is forwarded according to the QoS specifications set in the traffic policy. Packets that fail to meet any of the matching criteria are classified as members of the default traffic class.

2. **Create a traffic policy to associate the traffic class with one or more QoS features.**

   You use the **policy-map** *policy-map-name* global configuration command to create a traffic policy and to enter policy-map configuration mode. A traffic policy defines the QoS features to associate with the specified traffic class. A traffic policy contains three elements: a name; a traffic class (specified with the class policy-map configuration command); and the QoS policies configured in the class.

   - You name the traffic policy in the policy-map command line to enter policy-map configuration mode.
   - In policy-map configuration mode, enter the name of the traffic class used to classify traffic to the specified policy, and enter policy-map class configuration mode.
   - In policy-map class configuration mode, you can enter the QoS features to apply to the classified traffic. These include using the set, police, or police aggregate commands for input policy maps or the bandwidth, priority, queue-limit or shape average commands for output policy maps.

   Note

   A packet can match only one traffic class within a traffic policy. If a packet matches more than one traffic class in the traffic policy, the first traffic class defined in the policy is used. To configure more than one match criterion for packets, you can associate multiple traffic classes with a single traffic policy.

3. **Attach the traffic policy to an interface.**

   You use the **service-policy** interface configuration command to attach the policy map to an interface for packets entering or leaving the interface. You must specify whether the traffic policy characteristics should be applied to incoming or outgoing packets. For example, entering the service-policy output class1 interface configuration command attaches all the characteristics of the traffic policy named class1 to the specified interface. All packets leaving the specified interface are evaluated according to the criteria specified in the traffic policy named class1.

   Note

   If you enter the no policy-map configuration command or the no policy-map policy-map-name global configuration command to delete a policy map that is attached to an interface, the policy map is then detached and deleted.

Input and output policies

Explains policy maps that perform different functions based on whether they handle traffic entering or leaving the switch.

Input and output policies are policy map types that

- attach to packets as they enter or leave the switch through service policies applied to interfaces
- perform different traffic control functions depending on their position in the traffic flow, and
- share the same basic structure while regulating different traffic characteristics.

Policy map functions and limitations

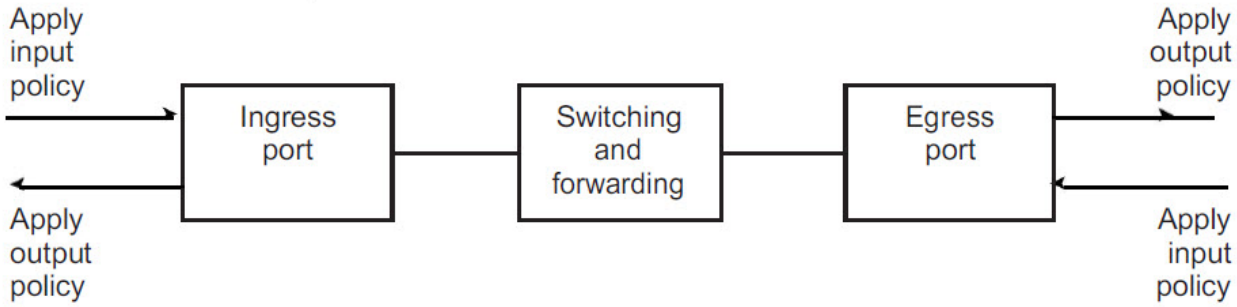
Input policy maps and output policy maps perform distinct functions:

- **Input policy maps**: Perform policing and marking on received traffic. Policed packets can be dropped or reduced in priority (marked down) if they exceed the maximum permitted rates.

- **Output policy maps**: Perform scheduling and queuing on traffic as it leaves the switch.

You can configure a maximum of 200 policy maps.

Figure 1: Input and output policy relationship



Input policy maps

Explains policy maps that classify traffic on input interfaces using CoS, DSCP, or ACL matching criteria.

An input policy map is a traffic classification mechanism that

- classifies traffic using CoS, DSCP, or access control list (ACL) matching criteria,
- applies actions such as setting or marking CoS or DSCP values and policing, and
- supports a maximum of 11 classes plus class-default.

Input policy map characteristics

Input policy maps support these actions:

- Setting or marking a CoS, or DSCP
- Policing

The class **class-default** is used in a policy map for any traffic that does not explicitly match any other class in the policy map. Input policy maps do not support queuing and scheduling keywords, such as **bandwidth**, **queue-limit**, **priority**, and **shape average**.

An input policy map can have a maximum of 11 classes plus class-default. You can configure a maximum of 11 classes in an input policy.

Output policy maps

Describes policy maps that control outbound traffic queuing, scheduling, and classification on switch ports.

An output policy map is a traffic management configuration that

- classifies outbound traffic based on CoS or DSCP values,
- applies queuing actions using **queue-limit**

  ,
- applies scheduling actions using **bandwidth**, **priority**, and **shape average**, and
- supports up to eight classes per policy map including class-default.

Output policy map characteristics

Output policy maps have specific capabilities and limitations:

- Do not support matching of access groups
- Do not support marking or policing except in the case of priority with policing
- Do not support egress packet marking (no set command in an output policy)

The class *class-default* is used in a policy map for any traffic that does not explicitly match any other class in the policy map. There can be a maximum of eight classes in the output policy map (including class-default) because egress ports have a maximum of eight queues.

You can attach an output policy map to any port on the switch. Each port can have a unique output policy map. However, each output policy map can contain only three unique queue-limit configurations. These three queue-limit configurations can be used in multiple output policy maps. There are no restrictions on bandwidth, priority, or shaping configurations.

Classification

Explains how QoS classification distinguishes different types of network traffic by examining packet header fields.

Classification is a QoS mechanism that

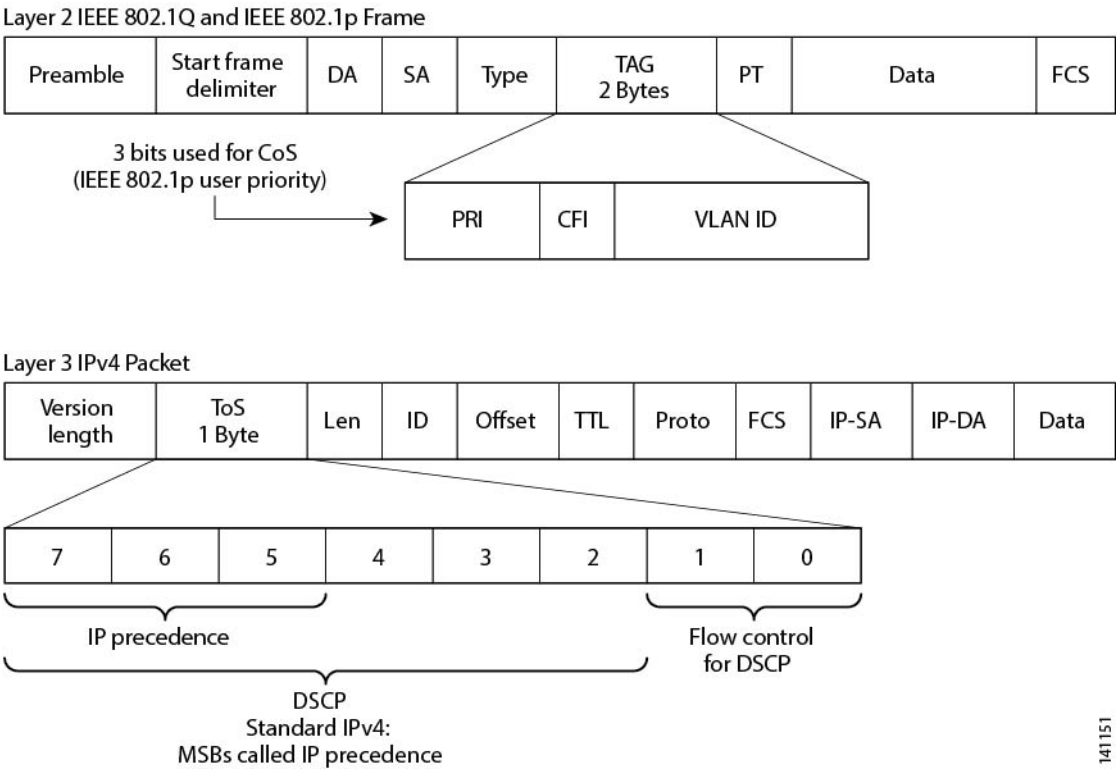
- distinguishes one kind of traffic from another by examining the fields in the packet header
- examines the header and identifies all key packet fields when a packet is received, and
- classifies packets based on an ACL, on the DSCP, or CoS.

QoS classification methods

Classification supports different methods for Layer 2 and Layer 3 packet processing:

- On ports configured as Layer 2 IEEE 802.1Q trunks, all traffic is in 802.1Q frames except for traffic in the native VLAN. Layer 2 802.1Q frame headers have a 2-byte tag control information field that carries the CoS value, called the user priority bits, in the three most-significant bits, and the VLAN ID value in the 12 least-significant bits. Other frame types cannot carry Layer 2 CoS values.
- Layer 2 CoS values range from 0 to 7.
- Both IPv4 and IPv6 packets can carry a DSCP value. QoS supports only the use of DSCP values.
- Use match on DSCP to classify IPv6 packets.

Figure 2: QoS classification layers in frames and packets



These sections contain additional information about classification:

Class maps

Match command

Configure classification based on layer 2 CoS

Classification based on IP DSCP

Classification comparisions

Classification based on QoS ACLs

Class maps

Explains how class maps define criteria to match and classify specific traffic flows for policy actions.

A class-map is an MQC configuration element that

- names a specific traffic flow or class and isolates it from all other traffic
- defines the criteria used to match against a specific traffic flow to further classify it, and
- enables packet matching against class-map criteria for action by the associated policy map.

Class-map configuration and matching

When you enter the **class-map** command with a class-map name, the switch enters the class-map configuration mode. In this mode, you define the match criterion for the traffic by using the match class-map configuration command. After a packet is matched against the class-map criteria, it is acted on by the associated action specified in a policy map.

You can match more than one criterion for classification. You can use the class-map match-any class-map name global configuration command to define a classification with any of the listed criteria.

If you have more than one type of traffic that you want to classify, you can create another class map and use a different name.

Note

If you do not enter match-all or match-any, the default is to match all. A match-all class map cannot have more than one classification criterion (match statement). A class-map with no match condition has a default of match all.

Match command

Describes the configuration of packet classification criteria within class-map configuration mode. This command enables network administrators to define specific traffic classes based on CoS, IP DSCP values, or access groups.

To configure the type of content used to classify packets, you use the **match class-map** configuration command to specify the classification criteria. If a packet matches the configured criteria, the device assigns it to a specific class and forwards it according to the specified policy. For example, you can use the **match class-map** command with CoS and IP DSCP values. These values are referred to as markings on a packet. You can also match an access group.

- In an input policy map, you cannot mix policy-map and class-map to classify IP and non-IP packets within the same policy-map.
- In an output policy map, no two class maps can have the same match criteria. For example, two different class maps cannot define a class that matches the same DSCP values.

Match command configuration

This example shows how to create a class map that matches any of the listed criteria. If a packet is received with DSCP equal to 32 or 40, the class map identifies (classifies) the packet.

```
Switch(config)# class-map match-any example
Switch(config-cmap)# match ip dscp 32
Switch(config-cmap)# match ip dscp 40
Switch(config-cmap)# exit
```

Classification based on IP DSCP

Describes IPv4 traffic classification methods using IP DSCP values with the match IP DSCP class-map configuration command.

Classification based on IP DSCP is a traffic management method that

- uses the **match ip dscp** class-map configuration command to classify IPv4 traffic based on DSCP values
- supports multiple classification options including specific DSCP values, service classes, and per-hop behaviors, and
- enables you to implement QoS policies by categorizing packets according to their DSCP markings.

Classification options

Available classification options display

When you classify IPv4 traffic based on IP DSCP value, you have several classification options:

- Entering a specific DSCP value (0 to 63).
- Using the default service, which corresponds to an IP precedence and DSCP value of 0. The default per-hop behavior (PHB) is usually best-effort service.

- Using Assured Forwarding (AF) by entering the binary representation of the DSCP value. AF sets the relative probability that a specific class of packets is forwarded when congestion occurs and the traffic does not exceed the maximum permitted rate. AF per-hop behavior provides delivery of IP packets in four different AF classes: AF11-13, AF21-23, AF31-33, and AF41-43. Each AF class could be allocated a specific amount of buffer space and drop probabilities, specified by the binary form of the DSCP number. When congestion occurs, the drop precedence of a packet determines the relative importance of the packet within the class. An AF41 provides the best probability of a packet being forwarded from one end of the network to the other.
- Entering Class Selector (CS) service values of 1 to 7, corresponding to IP precedence bits in the ToS field of the packet.
- Using Expedited Forwarding (EF) to specify a low-latency path. This corresponds to a DSCP value of 46. EF services use priority queuing to preempt lower priority traffic classes.

This display shows the available classification options:

```
Switch(config-cmap)# match ip dscp ?
<0-63> Differentiated services codepoint value
af11 Match packets with AF11 dscp (001010)
af12 Match packets with AF12 dscp (001100)
af13 Match packets with AF13 dscp (001110)
af21 Match packets with AF21 dscp (010010)
af22 Match packets with AF22 dscp (010100)
af23 Match packets with AF23 dscp (010110)
af31 Match packets with AF31 dscp (011010)
af32 Match packets with AF32 dscp (011100)
af33 Match packets with AF33 dscp (011110)
af41 Match packets with AF41 dscp (100010)
af42 Match packets with AF42 dscp (100100)
af43 Match packets with AF43 dscp (100110)
cs1 Match packets with CS1(precedence 1) dscp (001000)
cs2 Match packets with CS2(precedence 2) dscp (010000)
cs3 Match packets with CS3(precedence 3) dscp (011000)
cs4 Match packets with CS4(precedence 4) dscp (100000)
cs5 Match packets with CS5(precedence 5) dscp (101000)
cs6 Match packets with CS6(precedence 6) dscp (110000)
cs7 Match packets with CS7(precedence 7) dscp (111000)
default Match packets with default dscp (000000)
ef Match packets with EF dscp (101110)
```

For more information on DSCP prioritization, see RFC 2597 (AF per-hop behavior), RFC 2598 (EF), or RFC 2475 (DSCP).

Classification comparisons

Provides suggested IP DSCP, IP precedence, and CoS values for typical traffic types.

Table 1: Typical traffic classifications on page 10 shows suggested IP DSCP, IP precedence, and CoS values for typical traffic types.

Table 1: Typical traffic classifications

Traffic Type	DSCP per-hop	DSCP (decimal)	IP Precedence	CoS
Voice-bearer: Traffic is in a priority queue or in the queue with the highest service weight and lowest drop priority.	EF	46	5	5
Voice control: Signaling traffic related to call setup originates from a voice gateway or a voice application server.	AF31	26	3	3

Traffic Type	DSCP per-hop	DSCP (decimal)	IP Precedence	QoS
Video conferencing: In most networks, video conferencing over IP has similar loss, delay, and delay variation requirements as voice over IP traffic.	AF41	34	4	4
Streaming video: These are relatively high-bandwidth applications with a high tolerance for loss, delay, and delay variation. They are usually considered more important than regular background applications such as e-mail and web browsing.	AF13	14	1	1
Mission-critical data (gold data): Delay-sensitive applications that are critical to the operation of an enterprise.	-	-	-	-
	-	-	-	-
Level 1	AF21	18	2	2
Level 2	AF22	20	2	2
Level 3	AF23	22	2	2
Less-critical data (silver data): Noncritical but relatively important data.	-	-	-	-
	-	-	-	-
Level 1	AF11	10	1	1
Level 2	AF12	12	1	1
Level 3	AF13	14	1	1
Best-effort data (bronze data): Other traffic, including all default and non-interactive traffic, regardless of importance.	Default	0	0	0
Less-than-best-effort data: Noncritical, bandwidth-intensive data traffic that is given the least preference. This traffic type is the first to be dropped.	-	-	-	-
		-	-	-
Level 1		-	-	-
Level 2		2	0	0
Level 3		4	0	0
		6	0	0

Classification based on QoS ACLs

Explains how packets can be classified in input policy maps based on ACL lookup using IPv4, IPv6, or MAC ACLs with specific matching criteria and configuration requirements.

Classification based on QoS ACLs is a packet classification method that

- uses IPv4, IPv6, or MAC ACL lookup to classify packets in input policy maps,
- requires creating an ACL first, then configuring a class map with the match access-group command, and
- attaches the class map to a policy map to define traffic classification criteria.

Configuration requirements and limitations

Policy map configuration with class-maps

Classification based on QoS ACLs has specific configuration requirements and limitations:

- The maximum number of ACEs allowed under each ACL is 256. This number includes marking action rules configured in an input service policy and implicit rules.
- Only one access-group is supported per class for an input policy map

> **Note**
>
> You cannot configure match access-group for an output policy map.

You can use IP standard, IP extended, or Layer 2 MAC ACLs to define a group of packets with the same characteristics (a class). You use the **access-list** global configuration command to configure IP ACLs to classify IP traffic based on Layer 3 and Layer 4 parameters. You use the **mac access-list extended** global configuration command to configure Layer 2 MAC ACLs to classify IP and non-IP traffic based on Layer 2 parameters.

> **Note**
>
> You cannot match IP fragments against configured IP extended ACLs to enforce QoS. IP fragments are sent as best effort. IP fragments are denoted by fields in the IP header.

You can use only ACLs with a permit action in a match access-group command. ACLs with a deny action are never matched in a QoS policy.

In this example, the class maps in the policy map specify matching criteria for voice, data, and video traffic, and the policy map sets the action for input policing each type of traffic.

```
Switch(config)# policy-map policy-1
Switch(config-pmap)# class dscp-63
Switch(config-pmap-c)# police cir 10000000 bc 50000
Switch(config-pmap-c)# conform-action transmit
Switch(config-pmap-c)# exceed-action set-cos-transmit 5
Switch(config-pmap-c)# exit
Switch(config-pmap)# class dscp-1
Switch(config-pmap-c)# set cos 0
Switch(config-pmap-c)# exit
Switch(config-pmap)# class dscp-23
Switch(config-pmap-c)# set cos 4
Switch(config-pmap-c)# exit
```

Configure classification based on layer 2 CoS

Configure a class map to classify Layer 2 traffic based on the CoS value using the match command.

To classify Layer 2 traffic based on the Class of Service (CoS) value, which enables QoS traffic management and prioritization.

The CoS value ranges from 0 to 7 and is used to prioritize traffic on Layer 2 networks. You can use the **match** command to classify this traffic for policy application.

Before you begin

A **match cos** command is supported only on Layer 2 802.1Q trunk ports.

Follow these steps to configure classification based on Layer 2 CoS:

Procedure

1. Use the **class-map class-map-name** command to create a class map with a descriptive name and enter class-map configuration mode.

   Example

   ```
   Switch(config)# class-map premium
   ```

2. Use the **match cos cos-value** command to match traffic with the specified CoS value.

   Example

   ```
   Switch(config-cmap)# match cos 5
   ```

3. Use the **exit**command to return to global configuration mode.

   Example

   ```
   Switch(config-cmap)# exit
   ```

Policing

Describes a traffic regulation mechanism that limits bandwidth availability to specific traffic flows after packet classification.

Policing is a traffic regulation mechanism that

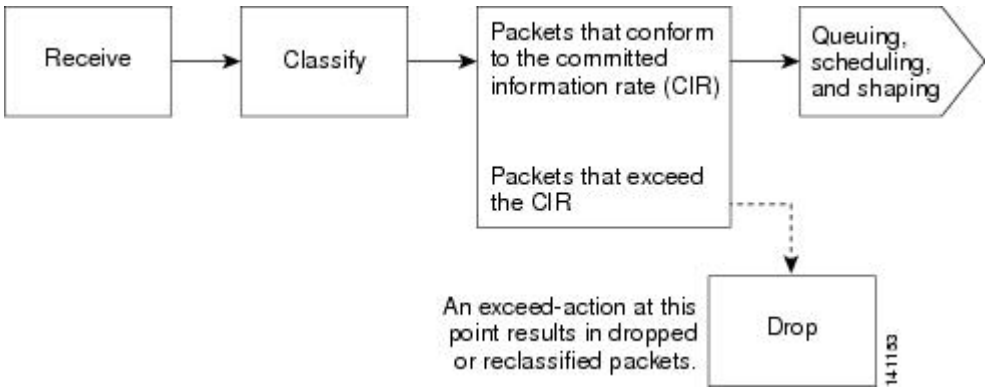
- limits the amount of bandwidth available to a specific traffic flow,
- prevents a traffic type from using excessive bandwidth and system resources, and
- identifies packets as in or out of profile by comparing the rate of inbound traffic to the configuration profile of the policer and traffic class.

Policing characteristics

Packets that exceed the permitted average rate or burst rate are out of profile or nonconforming. These packets are dropped or modified (marked for further processing), depending on the policer configuration.

Policing is used primarily on receiving interfaces. You can attach a policy map with a policer only in an input service policy. Policing is not allowed in an output policy map.

Figure 3: Policing of classified packets



Individual policing

Describes a quality of service (QoS) mechanism that applies rate limiting to input traffic on individual interfaces.

Individual policing is a QoS mechanism that

- applies only to input policy maps
- supports 1-rate, 2-color ingress policing and 2-rate, 3-color policing for individual or aggregate policing on Cisco Industrial Ethernet Switches, and
- defines committed rate limitations and burst size limitations for traffic classes with specified actions for conforming and non-conforming traffic.

Individual policing configuration and behavior

Multiple actions configuration

In policy-map configuration mode, you enter the **class** command followed by class-map name, and enter policy-map class configuration mode.

Note

There are two limitations to 'set' and 'police' in ingress policies:

- Set and police are not supported together in a single policy within a single class. However, if these actions are applied on different classes, then that is a supported configuration.
- The ASIC supports one marking action per policing configuration. On Cisco Catalyst IE3x00 Rugged Series switches, this marking action is available only for yellow packets.

For 1-rate, 2-color policing, you can use the **police** policy-map class configuration command to define the policer, the committed rate limitations of the traffic, committed burst size limitations of the traffic, and the action to take for a class of traffic that is below the limits (**conform-action**) and above the limits (**exceed-action**). If you do not specify burst size (bc), the system calculates an appropriate burst size value. The calculated value is appropriate for most applications.

When you configure a 2-rate policer, you must configure the committed information rate (CIR) for updating the first token bucket. You also configure the peak information rate (PIR) for updating the second token bucket. If you do not configure a PIR, the policer is a standard 1-rate, 2-color policer.

For 2-rate, 3-color policing, you can then optionally set actions to perform on packets that conform to the specified CIR and PIR (conform-action), packets that conform to the PIR, but not the CIR (exceed-action), and packets that exceed the PIR value (violate-action).

- If you set the CIR value equal to the PIR, a traffic rate that is less than or equal to the CIR is in the conform range. Traffic that exceeds the CIR is in the violate range.
- If you set the PIR greater than the CIR, a traffic rate less than the CIR is in the conform range. A traffic rate that exceeds the CIR but is less than or equal to the PIR is in the exceed range. A traffic rate that exceeds the PIR is in the violate range.
- If you do not configure a PIR, the policer is configured as a 1-rate, 2-color policer.

If burst sizes are set too low, throughput may be reduced in situations with bursty traffic. If burst sizes are set too high, the traffic rate may exceed acceptable limits.

Note

The switch supports byte counters for byte-level statistics for conform, exceed, and violate classes in the **show policy-map interface** privileged EXEC command output.

To make the policy map effective, you attach it to a physical port by using the **service-policy input** interface configuration command. Policing is done only on received traffic, so you can only attach a policer to an input service policy.

You can use the **conform-action** and **exceed-action** policy-map class configuration commands or the **conform-action** and **exceed-action** policy-map class police configuration commands to specify the action to take when the packet conforms to or exceeds the specified traffic rate.

Conform actions are to send the packet without modifications, or to drop the packet. Exceed actions are to drop the packet, to send the packet without modification, to set a new CoS, DSCP, or IP DSCP to a value.

You can use the **conform-action**, **exceed-action**, and **violate-action** policy-map class configuration commands or the **conform-action**, **exceed-action**, and **violate-action** policy-map class police configuration commands to specify the action to take when the packet conforms to or exceeds the specified traffic rates.

You can configure multiple conform and exceed actions simultaneously for each service class. You can configure multiple conform, exceed, and violate actions simultaneously for each service class.

To configure multiple actions in a class, enter multiple conform, exceed, or violate action entries in policy-map class police configuration mode.

```
Switch(config)# policy-map map1
Switch(config-pmap)# class class1
Switch(config-pmap-c)# police 100000 500000
Switch(config-pmap-c-police)# conform-action transmit
Switch(config-pmap-c-police)# exceed-action set-cos-transmit 2
Switch(config-pmap-c-police)# exit
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
```

Marking

Describes a QoS technique that sets or modifies packet attributes to identify specific traffic types for unique handling in network policy enforcement.

Marking is a QoS technique that

- sets or modifies packet attributes for traffic belonging to a specific class,
- identifies certain traffic types for unique handling after network traffic is organized into classes, and
- enables other QoS functions to judge the relative and absolute importance of packets.

Marking characteristics

Policy map marking configuration

You can use packet marking in input policy maps to change COS values in a class or set IP DSCP values for a specific type of traffic. These new values are then used to determine how the traffic should be treated.

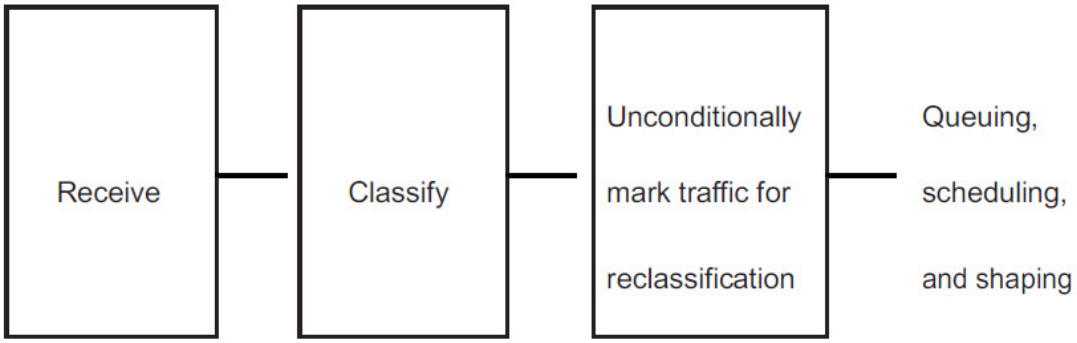
Traffic marking is typically performed on a specific traffic type at the ingress port. The marking action can cause the COS, DSCP, or precedence bits to be rewritten or left unchanged, depending on the configuration. Depending on the policy in the QoS domain, the priority of a packet can increase or decrease. Other QoS functions then use the marking information to judge the relative and absolute importance of the packet. The marking function can use information from the policing function or directly from the classification function.

You can specify and mark traffic by using the **set** commands in a policy map for all supported QoS markings (COS and IP DSCP). A **set** command unconditionally marks the packets that match a specific class. You then attach the policy map to an interface as an input policy map.

You can simultaneously configure actions to modify DSCP, precedence, and COS markings in the packet for the same actions.

This figure shows the steps for marking traffic.

Figure 4: Marking of classified traffic



This example uses a policy map to remark a packet. The first marking, using the set command, applies to the QoS default class map. This map matches all traffic not included in class AF31-AF33. It sets all such traffic to an IP DSCP value of 1. The second marking sets the traffic in classes AF31 to AF33 to an IP DSCP of 3.

```
Switch(config)# policy-map Example
Switch(config-pmap)# class class-default
Switch(config-pmap-c)# set ip dscp 1
Switch(config-pmap-c)# exit
Switch(config-pmap)# class AF31-AF33
Switch(config-pmap-c)# set ip dscp 3
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet1/1
Switch(config-if)# service-policy input Example
Switch(config-if)# exit
```

Congestion management and scheduling

Describes mechanisms that control outgoing traffic flow using Cisco Modular QoS CLI (MQC) to manage congestion and scheduling of packets.

Congestion management and scheduling is a traffic control mechanism that

- determines the order in which packets are sent out an interface based on priorities assigned to those packets
- creates queues and assigns packets to those queues based on packet classification, and
- schedules packets in a queue for transmission using various scheduling mechanisms.

Supported scheduling mechanisms

Cisco Modular QoS CLI (MQC) provides several related mechanisms to control outgoing traffic flow. They are implemented in output policy maps to control output traffic queues.

You can use different scheduling mechanisms to provide a guaranteed bandwidth to a particular class of traffic while also serving other traffic in a fair way. You can limit the maximum bandwidth consumed by a particular class of traffic. You can also ensure that delay-sensitive traffic in a low-latency queue is sent before traffic in other queues.

The switch supports these scheduling mechanisms:

- **Traffic shaping**: You use the **shape average** policy-map class command to configure average rate shaping. This command configures a maximum bandwidth for a particular class. The queue bandwidth is restricted to this value even though the port has more bandwidth available. You can configure shape average using either a percentage or a target bit rate value.
- **Class-based-weighted-fair-queuing (CBWFQ)**: You can use the **bandwidth** policy-map class configuration command to control the bandwidth allocated to a specific class. Minimum bandwidth can be specified as a percentage of total bandwidth or of remaining bandwidth.
  - **Bandwidth percent**: You can use the **bandwidth percent** policy-map class command to allocate a minimum bandwidth to a particular class. The total sum cannot exceed 100 percent. If the total sum is less than 100 percent, the rest of the bandwidth is divided equally among all bandwidth queues.
  - **Bandwidth Remaining**: You use the **bandwidth remaining** policy-map class command to create a ratio for sharing unused bandwidth in specified queues. Any unused bandwidth will be used by these specific queues in the ratio that is specified by the configuration. Use this command when the priority command is also used for certain queues in the policy.
- **Priority queuing or class-based priority queuing**: You use the **priority** policy-map class configuration command to specify the priority of a type of traffic over other types of traffic. You can specify strict priority for the known remaining bandwidth among the other traffic queues.
  - To configure strict priority, use only the **priority** policy-map class configuration command to configure the priority queue. Use the **bandwidth remaining percent** policy-map class configuration command for the other traffic classes to allocate the excess bandwidth in the desired ratios.
  - To configure priority with unconditional policing, configure the priority queue by using the **priority** policy-map class configuration command and the police policy-map class configuration command to unconditionally rate-limit the priority queue. In this case, you can configure the other traffic classes with **shape average**, depending on requirements.

Traffic shaping

Describes a traffic-control mechanism that manages the bandwidth of traffic leaving network interfaces to optimize data flow. It uses buffering to prevent packet loss during congestion.

Traffic shaping is a traffic-control mechanism similar to traffic policing. While traffic policing is used in input policy-maps, traffic shaping occurs as traffic leaves an interface. The switch can apply class-based shaping to classes of traffic leaving an interface and port shaping to all traffic leaving an interface. Configuring a queue for traffic shaping sets the maximum bandwidth or peak information rate, abbreviated as PIR of the queue.

Shaping uses a buffer to hold packets without enough tokens, preventing them from being immediately dropped. The number of buffers available to shaped traffic is limited and depends on various factors.

Note

Shape and priority cannot be configured within the same class in an output policy-map. However, shape and bandwidth can be configured together.

Class-based shaping

Describes a traffic management technique that limits data transmission rates for specific traffic classes using policy-map configuration commands.

Class-based shaping is a traffic management technique that

- uses the **shape average** policy-map class configuration command to limit the rate of data transmission as the number of bits per second to be used for the committed information rate for a class of traffic
- supports 8 queues per port on the transmit side, and
- designates the eighth queue as always default queue for class class-default, unclassified traffic.

Class-based shaping configuration

This example shows how to configure class-based shaping:

```
Switch(config-pmap)# class class-default
Switch(config-pmap-c)# shape average 90000000
Switch(config-pmap-c)# exit
Switch(config)# interface gigabitethernet1/1
Switch(config-if)# service-policy output out-policy
Switch(config-if)# exit
```

Class-based weighted fair queuing

Describes a bandwidth allocation method that sets relative queue precedence by allocating portions of available port bandwidth.

Class-based weighted fair queuing (CBWFQ) is a bandwidth allocation method that

- sets the relative precedence of a queue by allocating a portion of the total bandwidth available for the port
- uses the **bandwidth** policy-map class configuration command to set output bandwidth for a traffic class as a percentage of total bandwidth or remaining bandwidth, and
- provides minimum bandwidth guarantees or excess bandwidth allocation depending on the configuration method.

Bandwidth configuration methods

You can configure bandwidth allocation using two methods:

- **Percentage of total bandwidth**: When you use the **bandwidth** policy-map class configuration command to configure a class of traffic as a percentage of total bandwidth, this represents the minimum bandwidth guarantee (CIR) for that traffic class. This means that the traffic class gets at least the bandwidth indicated by the command, but is not limited to that bandwidth. Any excess bandwidth on the port is allocated to each class in the same ratio in which the CIR rates are configured.
- **Percentage of remaining bandwidth**: When you use the **bandwidth** policy-map class configuration command to configure a class of traffic as a percentage of remaining bandwidth, this represents the portion of the excess

bandwidth of the port that is allocated to the class. This means that the class is allocated bandwidth only if there is excess bandwidth on the port, and if there is no minimum bandwidth guarantee for this traffic class.

Configuration restrictions apply based on the method used:

- You cannot configure bandwidth as a percentage of total bandwidth when strict priority (priority without police) is configured for another class in the output policy.
- You can configure bandwidth as percentage of remaining bandwidth only when strict priority (priority without police) is configured for another class in the output policy map.

Note

You cannot configure bandwidth and traffic shaping (shape average) or priority queuing (priority) for the same class in an output policy map.

Note

When you configure CIR bandwidth for a class as a percentage of the total bandwidth, any excess bandwidth remaining after servicing the CIR of all the classes in the policy map is divided among the classes in the same proportion as the CIR rates. If the CIR rate of a class is configured as 0, that class is also not eligible for any excess bandwidth and as a result receives no bandwidth.

Priority queuing

Explains a traffic management mechanism that ensures certain classes of traffic receive preferential treatment over other traffic classes.

Priority queuing is a traffic management mechanism that

- ensures a particular class of traffic is given preferential treatment,
- allows traffic for the associated class to be sent before packets in other queues are sent, and
- constantly services the priority queue until all packets in the queue are scheduled and sent.

Priority queuing types and restrictions

Priority queuing configuration

The switch supports strict priority queuing, or you can use priority with the policy-map class submode command.

Strict priority queuing (priority without police) assigns a traffic class to a low-latency queue to ensure that packets in this class have the lowest possible latency. When this is configured, the priority queue is continually serviced until it is empty, possibly at the expense of packets in other queues.

> **Note**
>
> You should exercise care when using the **priority** command. Excessive use of strict priority queuing might cause congestion in other queues.

Priority queuing has these restrictions:

- You can associate the priority command with a single unique class for all attached output policies on the switch.
- You cannot configure priority and any other scheduling action (shape average or bandwidth) in the same class.
- You cannot configure priority queuing for the class-default of an output policy map.

This example shows how to configure the class out-class1 as a strict priority queue so that all packets in that class are sent before any other class of traffic. Other traffic queues are configured so that out-class-2 receives 50 percent of the remaining bandwidth, out-class-3 receives 20 percent, and the class-default queue receives the remaining 30 percent with no guarantees.

```
Switch(config)# policy-map policy1
Switch(config-pmap)# class out-class1
Switch(config-pmap-c)# priority
Switch(config-pmap-c)# exit
Switch(config-pmap)# class out-class2
Switch(config-pmap-c)# bandwidth remaining percent 50
Switch(config-pmap-c)# exit
Switch(config-pmap)# class out-class3
Switch(config-pmap-c)# bandwidth remaining percent 20
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
```

Congestion avoidance and queuing

Describes the network mechanisms that control packet flow and prevent network bottlenecks by managing queue sizes and implementing drop policies based on packet markings.

Congestion avoidance and queuing is a network traffic management mechanism that

- uses algorithms such as tail drop to control the number of packets entering the queuing and scheduling stage to avoid congestion and network bottlenecks
- assigns each packet to a specific queue and threshold based on markings such as DSCP or CoS values, and
- implements tail-drop mechanisms using drop precedence, buffer occupancy levels, and configured thresholds to decide whether to discard packets.

Tail drop mechanism operation

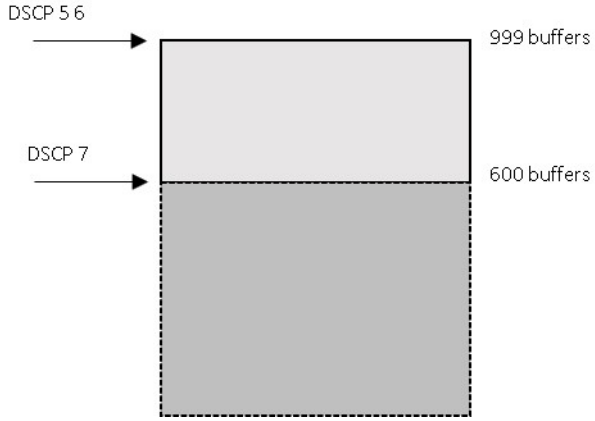
Tail drop with queue thresholds

Congestion avoidance selects the priority queue to which the packet is enqueued, and then implements the tail-drop mechanisms for individual queues. The tail-drop mechanism uses drop precedence associated with the packet, buffer occupancy levels, and configured thresholds to decide whether to discard the packet.

You set the queue size limits depending on the markings of the packets in the queue. You can specify the queue maximum threshold for the tail drop as number of packets. For example, specific DSCP or CoS values can be mapped to a specific egress queue and threshold.

This figure shows an example of tail drop operating on a queue with a maximum depth of 999 buffers. Two drop thresholds are allowed per queue. 999 buffers is the maximum for the queue. 600 buffers is the threshold setting for just DSCP 7, which is a subset of the classifiers matched by class "class_dscp_low". The default setting for the queue maximum is 1000 buffers. The maximum queue-limit can only be made smaller.

When the queue becomes congested and fills up with 600 or more buffers, and a new packet arrives marked with DSCP 7, the frame is subject to the 600 buffer limit threshold. The frame is tail dropped because the 600 buffer threshold would be exceeded.



The CLI for this example is shown below.

```
class-map match-any class_dscp_low
match dscp 5  6  7

policy-map p1
class class_dscp_low
bandwidth percent 50
queue-limit 999
queue-limit dscp 7 600
```

Queue limit configuration

Explains the configuration of queue limits using the queue-limit command to establish drop thresholds for traffic classes during congestion.

Queue limit configuration is a QoS mechanism that

- adjusts the queue size (buffer size) associated with a particular class of traffic using the **queue-limit** command in policy-map class sub mode
- establishes a drop threshold for the associated traffic when congestion occurs, and
- configures the threshold as the number of buffers (each buffer is a fixed unit of 256 bytes) or as a percentage for the traffic class.

Configuration requirements and restrictions

Queue limit configuration has specific requirements and restrictions:

Note

You cannot configure the queue size using the **queue-limit** command in policy-map class sub mode unless you first configure a scheduling action (bandwidth, shape average, or priority). The exception is when you set the queue-limit for the class-default of an output policy map.

The switch supports up to three unique queue-limit configurations across all output policy maps. Within any queue definition in an output policy map, you can define up to two thresholds. The first threshold is the maximum for the queue, which by default is 1000 buffers. The second threshold applies to a subset of classifiers (COS or DSCP). Multiple policy maps can share the same queue-limit values. If two policy maps share a queue-limit configuration, all threshold values for the classes in both policy maps must match.

You can use these same queue-limit values in multiple output policy maps on the switch. Changing any queue-limit value in a class creates a new, unique queue-limit configuration. You can attach only three unique queue-limit configurations in output policy maps to interfaces at any one time. If you attempt to attach an output policy map with a fourth unique queue-limit configuration, the system displays this error message:

```
QoS: Configuration failed. Maximum number of allowable unique queue-limit
configurations exceeded.
```

Note

When you configure a queue limit for a class in an output policy map, all other output policy maps must use the same match criteria for a class. Only the queue-limit threshold values can be different. For example, if you configure class A queue-limit thresholds for DSCP 30 and DSCP 50 in policy map PM1, you must use DSCP 30 and DSCP 50 as match criteria when configuring class A queue limits in policy map PM2. You cannot use DSCP 20 and DSCP 40. The threshold values may differ, but different thresholds create a new queue-limit configuration.

Buffer space allocation considerations:

- By default, the system divides the total buffer space equally among all ports and all queues per port. This allocation is adequate for most applications.
- You can decrease the queue size for latency-sensitive traffic or increase the queue size for bursty traffic.

Note

When you use the queue-limit command to configure queue thresholds for a class, the thresholds must be less than or equal to the queue maximum threshold.

When you configure queue limit, the range for the number of buffers is from 32 to 999, in multiples of 16, where each buffer is a fixed unit of 256 bytes.

> **Note**
>
> For optimal performance, we strongly recommend that you configure the queue-limit to 272 buffers or less.

Queue bandwidth and queue size (queue limit) configuration considerations:

- A larger queue limit can better accommodate bursty traffic without packet loss, but at the cost of increased latency.
- A smaller queue limit reduces latency but is more appropriate for steady traffic flows than for bursty traffic.
- Very small queue sizes are typically used to optimize priority queuing. For traffic that is priority queued, the buffer size usually needs to accommodate only a few packets. Large queue sizes that increase latency are not usually necessary. For high-priority, latency-sensitive packets, configure a relatively large bandwidth value and a relatively small queue size.

Restrictions apply to WTD qualifiers:

- You can configure only one threshold value for WTD qualifiers (COS, DSCP) by using the queue-limit command. However, there is no limit to the number of qualifiers that you can map to these thresholds. You can configure a second threshold value to set the maximum queue using the queue-limit command with no qualifiers.
- A WTD qualifier in the queue-limit command must be the same as at least one match criteria in the associated class map.

You can configure and attach as many output policy maps as there are switch ports, but only two unique queue-limit configurations are allowed. When another output policy map uses the same queue-limit and class configurations, even if the bandwidth percentages are different, it is considered to be the same queue-limit configuration.

Default QoS configuration

Describes the default QoS settings for traffic handling on egress ports. This configuration ensures traffic is processed without modification or policing.

The system operates with no pre-configured policy maps, class maps, or policers, ensuring traffic is switched in pass-through mode.

At the egress port, all traffic is directed to queues assigned a profile index based on CoS and DSCP values. Packets are not modified, meaning the CoS and DSCP values remain unchanged. Traffic is classified as best effort without any policing or rewrites.

Restrictions and limitations

Describes QoS configuration restrictions and operational limitations that apply to physical ports, traffic classification, and policy enforcement. These guidelines ensure stable network performance and prevent configuration errors.

This document provides the necessary configuration constraints and operational boundaries for implementing QoS features on the switch.

Configure QoS only on physical ports, as it is not supported on logical ports such as EtherChannel.

- You can configure QoS only on physical ports.
- On a port configured for QoS, all incoming traffic is classified, policed, and marked according to the input policy map attached to that port. For trunk ports, traffic in all VLANs is classified, policed, and marked based on the policy map applied to the port.
- QoS is not supported on logical ports (EtherChannel).

- All ingress QoS processing applies to control traffic—such as spanning-tree bridge protocol data units (BPDUs) and routing update packets—received by the switch.
- All control packets, such as CDP, LLDP, STP, and similar types, are marked for forwarding from queue 6 of egress ports.
- When you change queue settings, you are likely to lose data; therefore, try to make changes when traffic is at a minimum.
- When you attach a new policy and the number of policer instances exceeds 1024 minus the number of interfaces on the switch, an error message appears. The configuration does not complete.

Table 2: Port and interface configuration restrictions

Feature	Restriction or limits
QoS groups	Not supported.
Hierarchical QoS (HQoS)	Not supported.
Class maps	• A match-all class map cannot have more than one classification criterion (one match statement), but a match-any class map can contain multiple match statements. • The match cos command is supported only on Layer 2 802.1Q trunk ports. • For an input policy map, you cannot configure an IP classification (match ip dscp, match access-group for an IP ACL) and a non-IP classification (match cos or match access-group for a MAC ACL) in the same policy map or class map. • In an output policy-map, no two class maps can have the same classification criteria; that is, the same match qualifiers and values. • The maximum number of class maps on the switch is 1024.
Number of class maps in a policy map	11 plus class default for ingress policies, and 7 plus class default for egress policies.

Feature	Restriction or limits

ACLs for QoS	The EQ port comparator is not subject to these limits.
	• The EQ port comparator can be used in all ACEs up to the limit of ACEs in an ACL.
	• Port comparison actions other than EQ are LT, GT, and NEQ. These 3 port comparator types have limits. A total of 16 (8 comparators applied to TCP traffic and 8 comparators applied to UDP traffic) of these types of port comparators are allowed globally on the switch.
	• The switch supports IPv4 ACLs with range, lt, gt, and neq operations on TCP or UDP source or destination ports. Each range rule consumes 2 port-comparators.
	  • Number of Port comparators used for each per port comparison option: Range: Range of TCP or UDP ports - 2
	  • LT: “Lesser than” port # - 1
	  • GT: “Greater than” port # - 1
	  • NEQ: “Not Equal to” port # - 1
	• IPv6 ACLs with match on extension header and flow label are not supported.
	• IPv6 ACLs with match on object group and destination option are not supported.
	• IPv4 ACL with TTL based match is not supported.
	• IPv4 ACL with IP options based match is not supported.
	• Access groups cannot match on VLAN IDs.

Number of ACEs in a policy map	256 across all classes in a policy.
Match-all with multiple match criteria	Not supported.
Classmap matching on access-group of different types	Within a policy map, classmaps matching on access-group of different types is not supported. All class maps should be matching on the same type of access-group, (mac or ipv4 or ipv6), not a combination of these.
set and **police** in ingress policies	• Set and police are not supported together in a single policy within a single class. However, if these actions are applied on different classes, then that is a supported configuration. • The ASIC supports one marking action per policing configuration. On Cisco Catalyst IE3x00 Rugged Series switches, this marking action is available only for yellow packets.

Feature	Restriction or limits
Individual policers	• Policing is supported only on input policy maps. • The switch supports a maximum of 200 policers. • Policing is supported only on ingress ports. • The number of policer instances on the switch can be 1024 minus 1 more than the number interfaces. • The switch supports a maximum of 200 policer profiles. • If you do not configure a violate-action, by default the violate class is assigned the same action as the exceed-action.
`bandwidth qos-reference` *(value)*	No support for this interface level QoS command.
Policer and marking	This is supported only in the ingress direction.
Queueing and scheduling (bandwidth, shaper, priority, and queue-limit)	This is supported only in the egress direction.
Egress policy maps	Class maps in egress policy maps do not match on access-group.

QoS configuration

Explains the process of configuring Quality of Service (QoS) to classify, police, and mark incoming traffic, and schedule and queue outgoing traffic.

QoS configuration is a network management process that

- classifies, polices, and marks incoming traffic based on application types and traffic patterns
- schedules and queues outgoing traffic according to bandwidth requirements and congestion points, and
- requires thorough understanding of network characteristics before implementation.

Configuration requirements

Before configuring QoS, review Restrictions and limitations. You must also have a thorough understanding of these factors:

- The types of applications used and the traffic patterns on your network.
- The traffic characteristics and needs of your network. Is the traffic bursty? Do you need to reserve bandwidth for voice and video streams?
- Bandwidth requirements and speed of the network.
- Location of congestion points in the network.

These sections describe how to classify, police, and mark incoming traffic, and schedule and queue outgoing traffic.

Depending on your network configuration, you must perform one or more of these tasks:

- Use CoS and DSCP to classify traffic and Use ACLs to classify traffic
- Create IP extended ACLs on page 30 and Create layer 2 MAC ACLs on page 32
- Use class maps to define a traffic class on page 34

- Attach a traffic policy to an interface on page 36
- Best practice for configuring input policy-maps
- Configure input policy maps with individual policing on page 38 and Configure input policy maps with marking on page 43
- Output policy maps
- Configure output policy maps with class-based weighted fair queuing on page 46 and Configure output policy maps with port shaping on page 48
- Configure output policy maps with class-based priority queuing on page 49 and Configure output policy maps with weighted tail drop on page 52

Use CoS and DSCP to classify traffic

Configure class maps to classify traffic based on DSCP or CoS values for Quality of Service implementation.

Create class maps to define classes that match specific traffic criteria using DSCP or CoS values to classify packets for Quality of Service processing.

Traffic classification is essential for implementing Quality of Service policies. Class maps allow you to identify packets based on various criteria such as DSCP markings or CoS values, enabling appropriate treatment of different traffic types.

Procedure

1. Use the **class-map match-any** *class-map-name* command to create a class map that matches packets with DSCP values of 32 or 40.

   Example

   ```
   Switch(config)# class-map match-any example
   ```

2. Use the **match ip dscp** *dscp-value* command for each DSCP value that the class map must match.

   Example

   ```
   Switch(config-cmap)# match ip dscp 32
   Switch(config-cmap)# match ip dscp 40
   ```

   Because the class map uses the **match-any** option, the system classifies a packet if it has a DSCP value of 32 or 40.

3. Use the **exit** command to return to global configuration mode.
4. Use the **class-map** *class-map-name* command to create a class map for CoS-based classification and enter class-map configuration mode.

   Example

   ```
   Switch(config)# class-map class-cos
   ```

5. Use the **match cos** *cos-value* command to match packets with the specified CoS value.

Example

```
Switch(config-cmap)# match cos 2
```

6. Use the **exit** command to return to global configuration mode.

Example

```
Switch(config-cmap)# exit
```

Use ACLs to classify traffic

Describes how Access Control Lists (ACLs) can classify IP and non-IP traffic for network management purposes.

UsingACLs to classify traffic is a network management method that

- classifies IP traffic by using IP standard or IP extended Access Control Lists (ACLs),
- classifies IP and non-IP traffic by using Layer 2 MAC ACLs, and
- supports IPv4 ACLs with range, LT, GT, and NEQ operations on TCP or UDP source or destination ports.

ACL limitations and port comparators

The maximum number of Access Control Entries (ACEs) under every ACL is 256, including marking action rule configured in an input service policy and implicit rule.

The number of port comparators used for each per port comparison option is as follows:

- Range: Range of TCP or UDP ports–2
- LT: \"Lesser than\" port number–1
- GT: \"Greater than\" port number–1
- NEQ: \"Not equal to\" port number–1

Each range rule consumes 2 port-comparators. See Restrictions and limitations on page 23 for use of these port comparators.

Create IP standard ACLs

Configure IP standard ACLs for IP traffic using either numbered OR named access lists.

Create IP standard ACLs to control IP traffic by specifying which network traffic is permitted based on source addresses.

Use IP standard ACLs to filter traffic based on source IP addresses. These ACLs can use either numbered (1-99 OR 1300-1999) OR named configurations.

> **Note**
>
> Choose from Step 2 or Step 3 below.

Before you begin

Follow these steps to create an IP standard ACL for IP traffic:

Procedure

1. Use the **configure terminal** to enter global configuration mode.

   Example

   ```
   Switch# configure terminal
   ```

2. Use the **access-list** *access-list-number* **permit** *source [source-wildcard]* command to create an IP standard ACL.
   - For *access-list-number*, enter the access list number. The range is 1 to 99 and 1300 to 1999.
   - Always use the **permit** keyword for ACLs used as match criteria in QoS policies. QoS policies do not match ACLs that use the **deny** keyword.
   - For source, enter the network or host from which the packet is being sent. You can use the **any** keyword as an abbreviation for 0.0.0.0 255.255.255.255.
   - (Optional) For *source-wildcard*, enter the wildcard bits in dotted decimal notation to be applied to the source.

   Note

   The **show ip access-list** does not display hit counts OR matches at the software level, even when an ACL entry is matched on the following Cisco platforms and versions:

   Cisco Industrial Ethernet 9000 Series Switches

   Cisco Catalyst 9000 Series Switches

   Cisco Industrial Ethernet 3400 Series Switches

3. Use the **ip access-list standard** *name* command to create a named standard IPv4 access list and enter standard access-list configuration mode.

   Example

   ```
   Switch(config)# ip access-list standard qos-traffic
   ```

   Define a standard IPv4 access list using a name, and enter access-list configuration mode. The name can be a number from 1 to 99.

   In access-list configuration mode, enter **permit** source *[source-wildcard]*

4. Use the **end** command to return to privileged EXEC mode.

   Example

   ```
   Switch(config)# end
   ```

5. Use the **show access-lists** command to verify your entries.

   Example

   ```
   Switch# show access-lists
   ```

6. (Optional) Use the **copy running-config startup-config** command to save your entries in the startup configuration file.

Example

```
Switch# copy running-config startup-config
```

What to do next

To delete an access list, use the **no** access-list access-list-number global configuration command.

This example shows how to allow access for only those hosts on the three specified networks. The wild card bits apply to the host portions of the network addresses.

```
Switch(config)# access-list 1 permit 192.5.255.0 0.0.0.255
Switch(config)# access-list 1 permit 128.88.0.0 0.0.255.255
Switch(config)# access-list 1 permit 36.0.0.0 0.0.0.255
```

Create IP extended ACLs

Configure IP extended access control lists (ACLs) to filter network traffic based on various criteria including source, destination, protocol, and quality of service parameters.

Create IP extended ACLs to control network traffic flow and implement quality of service policies by filtering packets based on specific criteria such as source and destination addresses, protocols, and service parameters.

IP extended ACLs provide granular control over network traffic by allowing you to specify detailed filtering criteria. These ACLs are particularly useful for implementing QoS policies and traffic management strategies.

Beginning in privileged EXEC mode, follow these steps to create an IP extended ACL for IP traffic:

Procedure

---

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Switch# configure terminal
   ```

2. Use the **access-list** *access-list-number* **permit protocol** { **source** [*source-wildcard*] | **host source** | **any** } {**destination**[*destination-wildcard*] *precedence]* | **host destination** | **any**} [**precedence** *precedence*] [**tos** *tos*] **dscp** *dscp-value* command to create a numbered extended IPv4 ACL entry that permits packets matching the specified protocol, source and destination addresses, and IP precedence, ToS, or DSCP criteria.

   Create an IP extended ACL. Repeat the step as many times as necessary.

   > **Note**
   >
   > If you enter a `DSCP` value, you cannot enter `tos` or `precedence`. You can enter both a `tos` and a `precedence` value with no `DSCP`.

   - For *access-list-number*, enter the access list number. The range is 100 to 199 and 2000 to 2699.

- Always use the **permit** keyword for ACLs used as match criteria in QoS policies. QoS policies do not match **deny** ACLs.
- For source, enter the network or host from which the packet is being sent. You can use the any keyword as an abbreviation for 0.0.0.0 255.255.255.255.
- For *protocol*, enter the name or number of an IP protocol. Use the question mark (?) to display the available protocol values. To match any Internet protocol (including ICMP, TCP, and UDP), enter IP.
- The *source* is the number of the network or host sending the packet.
- The *source-wildcard* applies wildcard bits to the source.
- The *destination* is the network or host number receiving the packet.
- *The destination-wildcard* applies wildcard bits to the destination.

You can specify source, destination, and wildcards as:

- The 32-bit quantity in dotted-decimal format.
- The keyword **any** for 0.0.0.0 255.255.255.255 (any host).
- The keyword **host** for a single host 0.0.0.0.

Other keywords are optional and have these meanings:

- **precedence**–Enter to match packets with a precedence level specified as a number from 0 to 7 or by name: routine (0), priority (1), immediate (2), flash (3), flash-override (4), critical (5), internet (6), network (7).
- **tos**–Enter to match by type of service level, specified by a number from 0 to 15 or a name: normal (0), max-reliability (2), max-throughput (4), min-delay (8).
- **DSCP**–Enter to match packets with the DSCP value specified by a number from 0 to 63, or use the question mark (?) to see a list of available values.

3. Use the **ip access-list extended** *name* command to create a named extended IPv4 ACL and enter access-list configuration mode.

   Define an extended IPv4 access list using a name, and enter access-list configuration mode. The name can be a number from 100 to 199.

   In access-list configuration mode, enter **permit** *protocol {source-wildcard destination destination-wildcard}* **precedence** *precedence* **tos** *tos-value* DSCP *DSCP* as defined in the previous step.

4. Use the **end** command to return to privileged EXEC mode.

   Example

   ```
   Switch(config)# end
   ```

5. Use the **show access-lists** command to verify your entries.

   Example

   ```
   Switch# show access-lists
   ```

6. (Optional) Use the **copy running-config startup-config** command to save your entries in the startup configuration file.

What to do next

To delete an access list, use the no access-list access-list-number global configuration command.

This example shows how to create an ACL that permits IP traffic from any source to any destination that has the DSCP value set to 32:

```
Switch(config)# access-list 100 permit ip any any dscp 32
```

This example shows how to create an ACL that permits IP traffic from a source host at 10.1.1.1 to a destination host at 10.1.1.2 with a precedence value of 5:

```
Switch(config)# access-list 100 permit ip host 10.1.1.1 host 10.1.1.2 precedence
 5
```

This example shows how to create access list 103 and configure the class map called class1. The class1 has one match criterion, which is access list 103. It permits traffic from any host to any destination and includes range, lt, gt operator with specific source/destination port:

```
Switch(config)# access-list 103 permit udp any any lt 102
Switch(config)# access-list 103 permit tcp any any gt 1024
Switch(config)# access-list 103 permit tcp any any range 5555 5560
Switch(config)# access-list 103 permit udp any any range 2327 2499

Switch(config)# class-map match-any class1
Switch(config-cmap)# match access-group 103
Switch(config-cmap)#exit

Switch(config)# policy-map child
Switch(config-pmap)# class 1
Switch(config-pmap-c)# set ip dscp 1
Switch(config-pmap-c)# exit
```

Create layer 2 MAC ACLs

Configure Layer 2 MAC access control lists for non-IP traffic to control network access based on MAC addresses.

Layer 2 MAC ACLs allow you to control network access for non-IP traffic based on MAC addresses, providing granular security control at the data link layer.

Layer 2 MAC ACLs filter traffic based on source and destination MAC addresses and can include Ethertype specifications for additional protocol control.

Before you begin

Follow these steps to create a Layer 2 MAC ACL for non-IP traffic:

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Switch# configure terminal
   ```

2. Use the **mac access-list extended** *name* command to create a Layer 2 MAC ACL and enter extended MAC ACL configuration mode.

   Example

   ```
   Switch(config)# mac access-list extended name
   ```

3. Use the **permit** *{host src-MAC-addr mask | any | host dst-MAC-addr | dst-MAC-addr mask}[type mask]* command to add an entry that permits packets matching the specified source MAC address, destination MAC address, and optional EtherType criteria.

   Always use the permit keyword for ACLs used as match criteria in QoS policies.

   - For *src-MAC-addr*, enter the MAC address of the host from which the packet is being sent. You can specify in hexadecimal format (H.H.H), use the any keyword for source 0.0.0, source-wildcard ffff.ffff.ffff, or use the host keyword for source 0.0.0.
   - For *mask*, enter the wildcard bits by placing ones in the bit positions that you want to ignore.
   - For *dst-MAC-addr*, enter the MAC address of the host to which the packet is being sent. You can specify in hexadecimal format (H.H.H), use the any keyword for source 0.0.0, source-wildcard ffff.ffff.ffff, or use the host keyword for source 0.0.0.
   - (Optional) For *type mask*, specify the Ethertype number of a packet with Ethernet II or SNAP encapsulation to identify the protocol of the packet. For *type*, the range is from 0 to 65535, typically specified in hexadecimal. For *mask*, enter the don't care bits applied to the Ethertype before testing for a match.

4. Use the **end** command to return to privileged EXEC mode.

   Example

   ```
   Switch(config-ext-macl)# end
   ```

5. Use the **show access-lists** [*access-list-number | access-list-name*] command to verify your entries.

   Example

   ```
   Switch# show access-lists [access-list-number | access-list-name]
   ```

6. Use the **copy running-config startup-config** command to save your entries in the startup configuration file.

   Example

   ```
   Switch# copy running-config startup-config
   ```

What to do next

To delete an access list, use the **no mac access-list extended access** *list-name* global configuration command.

This example shows how to create a Layer 2 MAC ACL with two permit statements. The first statement allows traffic from the host with MAC address 0001.0000.0001 to the host with MAC address 0002.0000.0001. The second statement

allows only Ethertype XNS-IDP traffic from the host with MAC address 0001.0000.0002 to the host with MAC address 0002.0000.0002.

```
Switch(config)# mac access-list extended maclist1
Switch(config-ext-macl)# permit 0001.0000.0001 0.0.0 0002.0000.0001 0.0.0
Switch(config-ext-macl)# permit 0001.0000.0002 0.0.0 0002.0000.0002 0.0.0
xns-idp
Switch(config-ext-macl)# exit
```

Use class maps to define a traffic class

Configure class maps to name AND isolate specific traffic flows from all other traffic using match statements.

Use class maps to name and isolate specific traffic flows (or classes) from all other traffic. Class maps define the criteria to use to match against a specific traffic flow to further classify it.

Use the **class-map** global configuration command to name and to isolate a specific traffic flow (or class) from all other traffic. A class map defines the criteria to use to match against a specific traffic flow to further classify it. Match statements can include criteria such as an ACL, CoS value AND DSCP values. You define match criterion with one or more match statements entered in the class-map configuration mode.

Follow these guidelines when configuring class maps:

- A match-all class map cannot have more than one classification criterion (one match statement), but a match-any class map can contain multiple match statements.
- The match cos command is supported only on Layer 2 802.1Q trunk ports.
- For an input policy map, you cannot configure an IP classification (match IP DSCP, match access-group for an IP ACL) and a non-IP classification (match cos or match access-group for a MAC ACL) in the same policy map or class map.
- In an output policy map, no two class maps can have the same classification criteria; that is, the same match qualifiers and values.
- The maximum number of class maps on the switch is 1024.

Before you begin

Start in privileged EXEC mode, follow these steps to create a Layer 2 MAC ACL for non-IP traffic:

Procedure

1. Use the **configure terminal** command to enter the global configuration mode.

   Example

   ```
   Switch# configure terminal
   ```

2. Use the **class-map [match-all | match-any]** *class-map-name* command to create a class map and enter class-map configuration mode.

   Example

   ```
   Switch(config)# class-map [match-all | match-any] class-map-name
   ```

   Create a class map, AND enter class-map configuration mode. By default, no class maps are defined.

- (Optional) Use the **match-all** keyword to perform a logical-AND of all matching statements under this class map. All match criteria in the class map must be matched.
- (Optional) Use the **match-any** keyword to perform a logical-OR of all matching statements under this class map. One or more match criteria must be matched.
- For class-map-name, specify the name of the class map.

If no matching statements are specified, the default is **match-all**.

> **Note**
>
> A match-all class map cannot have more than one classification criterion (match statement).

3. Use the **match** {**access-group** *acl-index-or-name* | **cos** *cos-list* | **ip dscp** *dscp-list*} command to define the ACL, CoS, or IPv4 DSCP match criterion used to classify traffic.

   Define the match criterion to classify traffic. By default, no match criterion is defined.

   Only one match type per class map is supported, AND only one ACL per class map is supported.

   - For **access-group** *acl-index-or-name*, specify the number or name of an ACL. Matching access groups is supported only in input policy maps.
   - For **cos** *cos-list*, enter a list of up to four CoS values in a single line to match against incoming packets. Separate each value with a space. You can enter multiple *cos-list* lines to match more than four CoS values. The range is 0 to 7.
   - For **ip dscp** *dscp-list*, enter a list of up to eight IPv4 DSCP values to match against incoming packets. Separate each value with a space. You can enter multiple *dscp-list* lines to match more than eight DSCP values. The numerical range is 0 to 63. You can also configure DSCP values in other forms.

4. Use the **end** command to return to privileged EXEC mode.

   Example

   ```
   Switch(config-cmap)# end
   ```

5. Use the **show class-map** command to verify your entries.

   Example

   ```
   Switch# show class-map
   ```

6. Use the **copy running-config startup-config** command to save your entries in the startup configuration file.

   Example

   ```
   Switch# copy running-config startup-config
   ```

What to do next

Use the **no** form of the appropriate command to delete an existing class map or remove a match criterion.

This example shows how to create access list 103 AND configure the class map called class1. The class1 has one match criterion, which is access list 103. It permits traffic from any host to any destination that matches a DSCP value of 10.

```
Switch(config)# access-list 103 permit ip any any dscp 10
Switch(config)# class-map class1
Switch(config-cmap)# match access-group 103
Switch(config-cmap)# exit
```

This example shows how to create a class map called class2, which matches incoming traffic with DSCP values of 10, 11, AND 12.

```
Switch(config)# class-map match-any class2
Switch(config-cmap)# match ip dscp 10 11 12
Switch(config-cmap)# exit
```

Attach a traffic policy to an interface

Configure a traffic policy on a physical interface to control incoming or outgoing traffic using the service-policy command.

Attach a traffic policy to an interface to control traffic flow and apply Quality of Service (QoS) features to incoming or outgoing traffic on a physical port.

Use the **service-policy** interface configuration command to attach a traffic policy to an interface and to specify the direction in which the policy should be applied: either an input policy map for incoming traffic or an output policy map for outgoing traffic. Input and output policy maps support different QoS features.

You can attach a service policy only to a physical port. You can attach only one input policy map and one output policy map per port.

Note

If you enter the **no policy-map** configuration command or the no **policy-map** *policy-map-name* global configuration command to delete a policy map that is attached to an interface, a warning message appears that lists any interfaces from which the policy map is being detached. The policy map is then detached and deleted.

For example:

```
Warning: Detaching Policy test1 from Interface GigabitEthernet1/17
```

Start in privileged EXEC mode. Then, perform these steps to attach a policy map to a port:

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Switch# configure terminal
   ```

2. Use the **interface** *interface-id* command to specify a physical port and enter interface configuration mode.

   Example

   ```
   Switch(config)# interface interface-id
   ```

   For *interface-id*, specify the physical port to which you want to apply the policy map.

3. Use the **service-policy {input | output}** *policy-map-name* command to apply the policy map to the interface.

   Specify **input** to apply the policy map to incoming traffic or **output** to apply it to outgoing traffic. For **policy-map-name**, specify the name of the policy map.

4. Use the **end** command to return to privileged EXEC mode.
5. Use the **show policy-map interface** *interface-id* command to verify the policy-map configuration.

   Example

   ```
   Switch# show policy-map interface GigabitEthernet1/1
   ```

6. (Optional) Use the **copy running-config startup-config** command to save the running configuration to the startup configuration.

   Example

   ```
   Switch# copy running-config startup-config
   ```

Best practice for configuring input policy-maps

Outlines guidelines for configuring input policy maps that regulate traffic entering the switch.

In an input policy, you can match CoS, DSCP, or ACLs. You can also configure individual policing, aggregate policing, or marking to a CoS or DSCP value.

- You can attach only one input policy map per port.
- The maximum number of policy maps configured on the switch is 193.
- The total number of configurable policer profiles on the switch is 193.
- The maximum number of classes in each input policy map is 7 plus class-default.
- The number of input policy maps that can be attached in a switch depends on the available hardware resources. If you attach an input policy map that exceeds the hardware resource limit, the configuration does not succeed.
- After attaching a single-level policy map to an interface with the **service-policy input** interface configuration command, you can modify the policy without detaching it. You may add or delete classification criteria, classes, or actions, or change the parameters of the configured actions, such as policers, rates, mapping, and marking.
- On an 802.1Q tunnel port, you can use only an input policy map with Layer 2 classification based on MAC ACLs to classify traffic. Input policy maps with Layer 3 classification or with Layer 2 classification based on CoS or VLAN ID are not supported on tunnel ports.
- Input policy maps support policing and marking, not scheduling or queuing. You cannot configure **bandwidth**, **priority**, **queue-limit**, or **shape average** in input policy maps.

Configure input policy maps with individual policing

Configure individual policers to define committed rate limitations, committed burst size limitations of traffic, and actions to take for a class of traffic.

This task allows you to configure individual policers on input policy maps to control traffic flow by setting rate and burst limitations and defining actions for different classes of traffic.

Use the **police** policy-map class configuration command to configure individual policers to define the committed rate limitations, committed burst size limitations of the traffic, and the action to take for a class of traffic.

Follow these guidelines when configuring individual policers:

- Policing is supported only on input policy maps.
- The switch supports a maximum of 200 policers.
- Policing is supported only on the ingress port.
- The number of policer instances on the switch can be 1024 minus 1 more than the number interfaces. The switch supports a maximum of 200 policer profiles.
- If you do not configure a violate-action, the violate class is assigned the same action as the exceed-action by default.

Follow these guidelines when configuring class maps:

- A match-all class map cannot have more than one classification criterion (one match statement), but a match-any class map can contain multiple match statements.
- The match cos command is supported only on Layer 2 802.1Q trunk ports.
- For an input policy map, you cannot configure an IP classification (match IP dscp, match access-group for an IP ACL) and a non-IP classification (match cos OR match access-group for a MAC ACL) in the same policy map or class map.
- In an output policy map, no two class maps can have the same classification criteria; that is, the same match qualifiers and values.
- The maximum number of class maps on the switch is 1024.

Before you begin

Beginning in privileged EXEC mode, follow these steps to create an input policy map with individual policing:

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Switch# configure terminal
   ```

2. Use the **policy-map** *policy-map-name* command to create a policy map and enter policy-map configuration mode.

   Example

   ```
   Switch(config)# policy-map policy-map-name
   ```

   For *policy-map-name*, specify the name of the policy map. By default, the policy map does not contain any class maps.

3. Use the **class** { *class-map-name* | **class-default**}command to select a traffic class and enter policy-map class configuration mode.

   Example

   ```
   Switch(config-pmap)# class { class-map-name | class-default }
   ```

   Specify *class-map-name* to select an existing class map. You must have previously created the class map by using the **class-map** global configuration command.

   Specify **class-default** to match all packets that the other traffic classes do not classify.

4. Use the **police** {*rate-bps* | **cir** *cir-bps*} [*burst-bytes* |**bc** *burst-bytes*] command to define a policer for the traffic class.

   Example

   ```
   Switch(config-pmap-c)# police { rate-bps | cir cir-bps } [ burst-bytes | bc
   burst-bytes ]
   ```

   Define a policer for the class of traffic.

   - For *rate-bps*, specify average traffic rate in bits per second (bps). The range is 64000 to 1000000000.
   - For **cir** *cir-bps*, specify a committed information rate (CIR) in bits per second (bps). The range is 500 to 1000000000.
   - For *burst-bytes* (optional), specify the normal burst size in bytes. The range is 64000 to 1000000000.
   - For **bc** *burst-bytes* (optional), specify the conformed burst (bc) or the number of acceptable burst bytes. The range is 64000 to 100000000.

5. Optionally, configure the actions for packets that conform to or exceed the CIR:

   - Use the **conform-action { drop | transmit }** command to specify whether to drop or transmit packets that conform to the CIR.

     ```
     Switch(config-pmap-c-police)# conform-action { drop | transmit }
     ```

   - Use the **exceed-action cos** command to assign a new CoS value or use a configured mapping table for packets that exceed the CIR.

     ```
     Switch(config-pmap-c-police)# exceed-action cos { cos-value | cos [table]
      | dscp [table] }
     ```

     Note

     For **cos-value**, specify the new CoS value assigned to the classified traffic. The range is 0 to 7.

   - Use the **exceed-action [ip] dscp** command to assign a new DSCP value or use a configured mapping table for packets that exceed the CIR.

     ```
     Switch(config-pmap-c-police)# exceed-action [ip] dscp { dscp-value | cos
     [table] | dscp [table] }
     ```

> **Note**
>
> For *dscp-value*, specify the new DSCP value assigned to the classified traffic. The range is 0–63.
>
> You can include one **exceed-action** in the same command string as the **police** command. Alternatively, press Enter after entering the **police** command to enter policy-map class police configuration mode, where you can configure multiple actions. You must configure an action when you use this mode.
>
> If you explicitly include **exceed-action drop** in the **police** command, enter policy-map class police configuration mode and use the **no exceed action drop** command to remove that action before configuring a different exceed action.

6. Use the **exit** command three times to return to global configuration mode.

   Example

   ```
   Switch(config-pmap-c-police)# exit
   Switch(config-pmap-c)# exit
   Switch(config-pmap)# exit
   ```

7. Use the **interface** *interface-id* command to specify the interface to which you want to attach the policy map and enter interface configuration mode.

   Example

   ```
   Switch(config)# interface interface-id
   ```

8. Use the **service-policy input** *policy-map-name* command to attach the policy map created in Step 2 to the ingress interface.

   Example

   ```
   Switch(config-if)# service-policy input policy-map-name
   ```

9. Use the **end** command to return to privileged EXEC mode.

   Example

   ```
   Switch(config-if)# end
   ```

Verification

Use the **show policy-map** [*policy-map-name* [**class** *class-map-name*]] command to verify your entries.

```
Switch# show policy-map [policy-map-name [class class-map-name]]
```

Optionally, use the **copy running-config startup-config** command to save your entries in the startup configuration file.

```
Switch# copy running-config startup-config
```

Configure input policy maps with individual 2-rate, 3-color policing

Configure input policy maps with individual 2-rate, 3-color policing to control traffic rates and define actions for conform, exceed, and violate conditions.

This task creates an input policy map with individual 2-rate, 3-color policing to control traffic flow by defining committed information rate (CIR) and peak information rate (PIR) parameters with corresponding actions for different traffic conditions.

Use 2-rate, 3-color policing to implement traffic rate limiting with more granular control than single-rate policing. This configuration allows you to define different actions for packets that conform to the CIR, exceed the CIR but are within the PIR, and violate both rates.

Follow these steps to create an input policy map with individual 2-rate, 3-color policing:

> **Note**
>
> In the procedure below, perform Step 5, 6, OR 7.

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Switch# configure terminal
   ```

2. Use the **policy-map** *policy-map-name* command to create a policy map and enter policy-map configuration mode.

   Example

   ```
   Switch(config)# policy-map policy-map-name
   ```

   For *policy-map-name*, specify the name of the policy map. By default, the policy map does not contain any class maps.

3. Use the **class** {*class-map-name* | **class-default**} command to select a traffic class and enter policy-map class configuration mode.

   Example

   ```
   Switch(config-pmap)# class { class-map-name | class-default }
   ```

   Specify *class-map-name* to select an existing class map. You must have previously created the class map by using the **class-map** global configuration command.

   Specify **class-default** to match all packets that the other traffic classes do not classify.

4. Use the **police** {*rate-bps* [*burst-bytes*] | **cir** *cir-bps* [**bc** *conform-burst*] [**pir** *pir-bps* [**be** *peak-burst*]]} command to define a one-rate or two-rate policer for the traffic class.

   - For *rate-bps*, specify average traffic rate in bits per second (bps). The range is 8000 to 1000000000.
   - For **CIR** *CIR-bps*, specify a committed information rate at which the bc token bucket is updated in bits per second (b/s). The range is 500 to 1000000000.
   - For *burst-bytes* (optional), specify the normal burst size in bytes. The range is 8000 to 1000000.

- For **bc** *burst-bytes* (optional), specify the conformed burst (bc) OR the number of acceptable burst bytes. The range is 64000 to 100000000.
- (Optional) For **bc** *conform-burst*, specify the conformed burst used by the bc token bucket for policing. The range is 8000 to 1000000 bytes.
- (Optional) For **PIR** *PIR-bps*, specify the peak information rate at which the be token bucket for policing is updated. The range is 500 to 1000000000 b/s. If you do not enter a **PIR** *PIR-bps*, the policer is configured as a 1-rate, 2-color policer.
- For **be** *peak-burst*, specify the peak burst size used by the be token bucket. The range is 8000 to 1000000 bytes. The default is internally calculated based on the user configuration.

5. Use the **conform-action [drop | transmit]** command to specify the action for packets that conform to the CIR.

   Example

   ```
   Switch(config-pmap-c-police)# conform-action { drop | transmit }
   ```

   Use the **exceed-action** command to specify the action for packets that exceed the CIR.

   ```
   Switch(config-pmap-c-police)# exceed-action { drop | set-cos-transmit {
   cos-value | cos } | set-dscp-transmit { dscp-value | dscp } }
   ```

   Use the **violate-action { drop | transmit }** command to specify the action for packets that violate the PIR.

   ```
   Switch(config-pmap-c-police)# violate-action { drop | transmit }
   ```

6. Use the **exit** command three times to return to global configuration mode..

   Example

   ```
   Switch(config-pmap-c-police)# exit
   Switch(config-pmap-c)# exit
   Switch(config-pmap)# exit
   ```

7. Use the **interface** *interface-id* command to specify the interface to which you want to attach the policy map and enter interface configuration mode.

   Example

   ```
   Switch(config)# interface interface-id
   ```

8. Use the **service-policy input** *policy-map-name* command to attach the policy map created in Step 2 to the ingress interface.

   Example

   ```
   Switch(config-if)# service-policy input policy-map-name
   ```

9. Use the **end** command to return to privileged EXEC mode.

   Example

   ```
   Switch(config-if)# end
   ```

Verification

Use the **show policy-map** [*policy-map-name* | **interface**] command to verify your entries.

```
Switch# show policy-map [policy-map-name | interface]
```

Optionally, use the **copy running-config startup-config** command to save your entries in the startup configuration file.

```
Switch# copy running-config startup-config
```

Configure input policy maps with marking

Configure input policy maps to mark traffic by setting new values in packets for quality of service classification.

Configure policy maps to set or modify attributes for traffic belonging to specific classes by marking packets with new CoS or DSCP values.

You use the **set policy-map class** configuration command to set or modify the attributes for traffic belonging to a specific class.

Before you begin

Follow these steps to create an input policy map that marks traffic:

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Switch# configure terminal
   ```

2. Use the **policy-map** *policy-map-name* command to create a policy map and enter policy-map configuration mode.

   Example

   ```
   Switch(config)# policy-map policy-map-name
   ```

   For **policy-map-name**, specify the name of the policy map.

3. Use the **class** {*class-map-name* | **class-default}** command to select a traffic class and enter policy-map class configuration mode.

   Example

   ```
   Switch(config-pmap)# class { class-map-name | class-default }
   ```

   Specify **class-map-name** to select an existing class map. You must have previously created the class map by using the **class-map** global configuration command. Specify **class-default** to match all packets that the other traffic classes do not classify.

4. Choose one of these commands to mark classified traffic by assigning a new value to the packet:
   - Use the **set cos** {*cos-value*} command to assign a new CoS value. The valid range is 0–7.

Example

```
Switch(config-pmap-c)# set cos 5
```

- Use the **set [IP] dscp** *dscp-value* command to assign a new DSCP value. The valid range is 0–63.

Example

```
Switch(config-pmap-c)# set ip dscp 6
```

5. Use the **exit** command to return to policy-map configuration mode.

Example

```
Switch(config-pmap-c)# exit
```

6. Use the **exit** command to return to global configuration mode.

Example

```
Switch(config-pmap)# exit
```

7. Use the **interface** *interface-id* command to attach the policy map created in Step 2 to the ingress interface.

Example

```
Switch(config)# interface interface-id
```

8. Use the **service-policy input** *policy-map-name* command to attach the policy map created in Step 2 to the ingress interface.

Example

```
Switch(config-if)# service-policy input policy-map-name
```

9. Use the **end** command to return to privileged EXEC mode.

Example

```
Switch(config-if)# end
```

Verification

Use the **show policy-map [policy-map-name** [*class class-map-name*]] command to verify your entries. Optionally, use the **copy running-config startup-config** command to save your entries in the startup configuration file.

What to do next

Use the **no** form of the appropriate command to delete a policy map or remove an assigned CoS or DSCP value.

This example uses a policy map to remark a packet. The first marking (the set command) applies to the QoS default class map that matches all traffic not matched by class AF31-AF33 and sets all traffic to an IP DSCP value of 1. The second marking sets the traffic in classes AF31 to AF33 to an IP DSCP of 3.

```
Switch(config)# policy-map Example
Switch(config-pmap)# class class-default
Switch(config-pmap-c)# set ip dscp 1
Switch(config-pmap-c)# exit
Switch(config-pmap)# class AF31-AF33
Switch(config-pmap-c)# set ip dscp 3
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface Gi1/3
Switch(config-if)# service-policy input Example
Switch(config-if)# exit
```

In this example, the class maps in the policy map specify matching criteria for voice, data, and video traffic, and the policy map sets the action for input policing each type of traffic.

```
Switch(config)# policy-map policy-1
Switch(config-pmap)# class dscp-63 voice
Switch(config-pmap-c)# police cir 10000000 bc 50000
Switch(config-pmap-c)# conform-action transmit
Switch(config-pmap-c)# exceed-action set-cos-transmit 5
Switch(config-pmap-c)# exit
Switch(config-pmap)# class dscp-1 data
Switch(config-pmap-c)# set cos 0
Switch(config-pmap-c)# exit
Switch(config-pmap)# class dscp-23 video
Switch(config-pmap-c)# set cos 4
Switch(config-pmap-c)# exit
```

Output policy maps

Explains the configuration and management of output policy maps used for congestion avoidance, queuing, and scheduling of packets leaving the switch.

An output policy map is a traffic management configuration that

- manages congestion avoidance, queuing, and scheduling of packets leaving the switch
- controls traffic on the four egress queues through shaping, queue-limit, and bandwidth configuration
- supports class-based priority queuing and uses CoS or DSCP for classification criteria.

Configuration guidelines

Output policy maps on physical ports have specific configuration requirements and limitations:

- Output policy maps can have a maximum of eight classes, including the class **class-default**.
- Class maps in an output policy map cannot use access-groups.
- All output policy maps must use the same set of classes, although the actions for each class can differ for each output policy map.
- You cannot configure class-based priority queuing under the class **class-default** in an output policy map.

- In an output policy map, unless priority queuing is configured, the class default receives a minimum bandwidth guarantee equal to the unconfigured bandwidth on the port.
- After attaching an output policy map to an interface with the **service-policy** interface configuration command, you can only change the parameters of already configured actions, such as rates and percentages, or adjust the classification criteria of the class map while the policy map remains attached. To add or remove a class or an action, detach the policy map from all interfaces, modify it, and then reattach it.
- We recommend configuring the interface to operate a fixed port speed when an output policy map depends on port bandwidth. Use the appropriate **interface-level** command to fix the port speed and disable auto-negotiation for speed; you may leave duplex negotiation enabled. If an output policy map is configured on a port set to auto-negotiate speed and the negotiated value invalidates the policy map, the port enters the error-disabled state
- You can attach only one output policy map per port.
- The maximum number of policy maps configured on the switch is 256.

If you expect to need three classes in a policy map, define all three when you create the policy map, even if you do not plan to use them immediately. Once attached to an interface, you cannot add additional classes to the policy map.

Configure output policy maps with class-based weighted fair queuing

Configure class-based weighted fair queuing (CBWFQ) to set the relative precedence of a queue by allocating a portion of the total bandwidth available for the port.

Configure class-based weighted fair queuing (CBWFQ) to control bandwidth allocated to a traffic class by specifying a minimum bandwidth as a bit rate or a percentage.

Use the **bandwidth** policy-map class configuration command to configure class-based weighted fair queuing (CBWFQ). CBWFQ sets the relative precedence of a queue by allocating a portion of the total bandwidth that is available for the port.

Follow these guidelines when configuring CBWFQ:

Cisco recommends configuring the interface to operate a fixed port speed if there is an output policy-map applied that has dependencies on port bandwith. Use the **interface level** command to fix the port speed, and remove the auto negotiation of speed. You can leave the negotiation of duplex. If an output policy-map is configured on a port that is set to auto negotiate speed, and the speed negotiates to a value rate that invalidates the output policy-map, the port is put into error-disabled state.

Before you begin

Follow these steps to use CBWFQ to control bandwidth allocated to a traffic class by specifying a minimum bandwidth as a bit rate or a percentage:

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Switch# configure terminal
   ```

2. Use the **policy-map** *policy-map-name* command to create a policy map and enter policy-map configuration mode.

   Example

   ```
   Switch(config)# policy-map policy-map-name
   ```

   For *policy-map-name*, specify the name of the policy map.

3. Use the **class** *{class-map-name* | **class-default}** command to select a child traffic class and enter policy-map class configuration mode.

   Example

   ```
   Switch(config-pmap)# class { class-map-name | class-default }
   ```

   Specify *class-map-name* to select a child class map or **class-default** to match all packets that the other traffic classes do not classify.

4. Use the **bandwidth** { *rate* | **percent** *value* | **remainingpercent** *value*} command to set an output bandwidth limit for the policy-map class.

   Example

   ```
   Switch(config-pmap-c)# bandwidth { rate | percent value | remaining value }
   ```

   Set output bandwidth limits for the policy-map class.

   - Enter a **rate** to set bandwidth in kilobits per second. The range is from 64000 to 1000000000.
   - Enter **percent** value to set bandwidth as a percentage of the total bandwidth. The range is 1 to 100 percent.
   - Enter **remaining percent** value to set bandwidth as a percentage of the remaining bandwidth. The range is 1 to 100 percent. This keyword is valid only when strict priority (priority without police) is configured for another class in the output policy map.

     You must specify the same units in each bandwidth configuration in an output policy (absolute rates or percentages). The total guaranteed bandwidth cannot exceed the total available rate.

5. Use the **exit** command twice to return first to policy-map configuration mode and then to global configuration mode.

   Example

   ```
   Switch(config-pmap-c)# exit
   Switch(config-pmap)# exit
   ```

6. Use the **interface** *interface-id* command to specify the interface to which you want to attach the policy map and enter interface configuration mode.

   Example

   ```
   Switch(config)# interface interface-id
   ```

7. Use the **service-policy output** *policy-map-name* command to attach the policy map created in Step 2 to the egress interface.

   Example

   ```
   Switch(config-if)# service-policy output policy-map-name
   ```

8. Use the **end** command to return to privileged EXEC mode.

Example

```
Switch(config-if)# end
```

Verification

Use the **show policy-map** [*policy-map-name* [**class** *class-map-name*]] command to verify your entries.

```
Switch# show policy-map [policy-map-name [class class-map-name]]
```

Optionally, use the **copy running-config startup-config** command to save your entries in the startup configuration file.

```
Switch# copy running-config startup-config
```

Configure output policy maps with port shaping

Configure port shaping to specify the maximum permitted average rate for a class of traffic using output policy maps.

Configure port shaping to control the maximum bandwidth for all traffic leaving an interface using a policy map with class default.

Port shaping is applied to all traffic leaving an interface. It uses a policy map with only class default when the maximum bandwidth for the port is specified by using the shape average command.

Before you begin

Start in privileged EXEC mode, follow these steps to use port shaping to configure the maximum permitted average rate for a class of traffic:

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Switch# configure terminal
   ```

2. Use the **policy-map** *policy-map-name* command to create a policy map and enter policy-map configuration mode.

   Example

   ```
   Switch(config)# policy-map policy-map-name
   ```

   For *policy-map-name*, specify the name of the policy map.

3. Use the **class class-default** command to select the default traffic class and enter policy-map class configuration mode.

   Example

   ```
   Switch(config-pmap)# class class-default
   ```

4. Use the **shape average** *target bps* command to configure the average class-based shaping rate.

Example

```
Switch(config-pmap-c)# shape average target-bps
```

Specify the average class-based shaping rate. For target bps, specify the average bit rate in bits per second. The range is from 64000 to 1000000000.

5. Use the **interface** *interface-id* command to specify the interface to which you want to attach the policy map and enter interface configuration mode.

   Example

```
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface GigabitEthernet1/1
```

6. Use the **service-policy output** *policy-map-name* command to attach the policy map created in Step 2 to the egress interface.

   Example

```
Switch(config-if)# service-policy output policy-map-name
```

7. Use the **end** command to return to privileged EXEC mode.

   Example

```
Switch(config-if)# end
```

8. Use the **show policy-map** [*policy-map-name* [**class** *class-map-name*]] command to verify your entries.

   Example

```
Switch# show policy-map [policy-map-name [class class-map-name]]
```

9. (Optional) Use the **copy running-config startup-config** command to save your entries in the startup configuration file.

   Example

```
Switch# copy running-config startup-config
```

Configure output policy maps with class-based priority queuing

Configure output policy maps with class-based priority queuing to ensure that a particular class of traffic is given preferential treatment over other traffic classes.

Configure output policy maps with class-based priority queuing to ensure that a particular class of traffic is given preferential treatment.

You can use the priority policy-map class configuration command to ensure that a particular class of traffic is given preferential treatment. With strict priority queuing, the priority queue is constantly serviced; all packets in the queue are

scheduled and sent until the queue is empty. Excessive use of the priority queues can possibly delay packets in other queues and create unnecessary congestion.

You can configure strict priority queuing (priority without police), or you can configure an unconditional priority policer (priority with police). Follow these guidelines when configuring priority queuing:

- You can associate the priority command with a single unique class for all attached output policies on the switch.
- When you configure a traffic class as a priority queue, you can configure only police and queue-limit as other queuing actions for the same class. You cannot configure bandwidth or shape average with priority queues in the same class.
- You cannot associate the priority command with the class-default of the output policy map.
- When you configure priority queuing without policing for a traffic class, you can only configure the other queues for sharing by using the bandwidth remaining percent policy-map class configuration command to allocate excess bandwidth. This command does not guarantee the allocated bandwidth, but does ensure the rate of distribution.

Before you begin

Follow these steps to configure a strict priority queue:

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Switch# configure terminal
   ```

2. Use the **class-map** *class-map-name* command to create a class map for each of the three egress queues.

   Example

   ```
   Switch(config)# class-map class-map-name
   ```

   Configure the applicable match conditions for each class. Repeat this step until you have created classes for all three egress queues.

3. Use the **policy-map** *policy-map-name* command to create a policy map and enter policy-map configuration mode.

   Example

   ```
   Switch(config)# policy-map policy-map-name
   ```

   For **policy-map-name**, specify the name of the policy map.

4. Configure strict scheduling priority for the priority class:
   - Use the **class** *class-map-name* command to select the priority class created in Step 2 and enter policy-map class configuration mode.

     ```
     Switch(config-pmap)# class class-map-name
     ```

   - Use the **priority** command to assign strict scheduling priority to the class.

     ```
     Switch(config-pmap-c)# priority
     ```

> **Note**
>
> Only one unique class map on the switch can be associated with the **priority** command. You cannot configure **priority** with another queuing action, such as **bandwidth** or **shape average**, in the same class.

- Use the **exit** command to return to policy-map configuration mode.

```
Switch(config-pmap-c)# exit
```

5. Configure bandwidth scheduling for each nonpriority class:
   - Use the **class** *class-map-name* command to select a nonpriority class and enter policy-map class configuration mode.

```
Switch(config-pmap)# class class-map-name
```

   - Use the **bandwidth remaining percent** *value* command to allocate a percentage of the remaining bandwidth to the class.

```
Switch(config-pmap-c)# bandwidth remaining percent value
```

> **Note**
>
> For **value**, specify a percentage from 1 through 100.

   - Use the **exit** command to return to global configuration mode.

```
Switch(config-pmap)# exit
```

6. Use the **interface** *interface-id* command to specify the interface to which you want to attach the policy map and enter interface configuration mode.

   Example

```
Switch(config)# interface interface-id
```

7. Use the **service-policy output** *policy-map-name* command to attach the policy map created in Step 3 to the egress interface.

   Example

```
Switch(config-if)# service-policy output policy-map-name
```

8. Use the **end** command to return to privileged EXEC mode.

Example

```
Switch(config-if)# end
```

Verification

Use the **show policy-map** command to verify your entries.

```
Switch# show policy-map [policy-map-name [class class-map-name]]
```

Optionally, use the **copy running-config startup-config** command to save your entries in the startup configuration file.

```
Switch# copy running-config startup-config
```

What to do next

After you have created an output policy map, you attach it to an egress port.

Use the no form of the appropriate command to delete an existing policy map or class map or to cancel strict priority queuing for the priority class or the bandwidth setting for the other classes.

This example shows how to configure the class out-class1 as a strict priority queue so that all packets in that class are sent before any other class of traffic. Other traffic queues are configured so that out-class-2 gets 50 percent of the remaining bandwidth and out-class3 gets 20 percent of the remaining bandwidth. The class class-default receives the remaining 30 percent with no guarantees.

```
Switch(config)# policy-map policy1
Switch(config-pmap)# class out-class1
Switch(config-pmap-c)# priority
Switch(config-pmap-c)# exit
Switch(config-pmap)# class out-class2
Switch(config-pmap-c)# bandwidth remaining percent 50
Switch(config-pmap-c)# exit
Switch(config-pmap)# class out-class3
Switch(config-pmap-c)# bandwidth remaining percent 20
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface GigabitEthernet1/17
Switch(config-if)# service-policy output policy1
Switch(config-if)# exit
```

Configure output policy maps with weighted tail drop

Configure weighted tail drop (WTD) to adjust the queue size associated with a traffic class using the queue-limit policy-map class configuration command.

Weighted tail drop (WTD) adjusts the queue size (buffer size) associated with a traffic class. You configure WTD by using the queue-limit policy-map class configuration command.

Follow these guidelines when configuring WTD:

- Configuring WTD with the queue-limit command is supported only when you first configure a scheduling action, such as bandwidth, shape average, or priority. The exception to this is when you are configuring queue-limit in the class-default.
- You can configure and attach as many output policy maps as there are ports, but you can attach only three unique queue-limit configurations in output policy maps to interfaces at any one time. Multiple output policy maps can use

the same queue-limit configuration. Changing one of the queue-limit values in a class creates a new, unique queue-limit configuration.

- When you configure a queue limit for a class in an output policy map, all other output policy maps must use the same match criteria for a class. Only the queue-limit threshold values can be different. For example, when you configure class A queue limit thresholds for DSCP 30 and DSCP 50 in policy map PM1, and you configure class A queue limits in policy map PM2, you must use DSCP 30 and DSCP 50 as qualifiers. You cannot use DSCP 20 and DSCP 40. The threshold values can be different, but different threshold values creates a new queue-limit configuration.
- When you use the queue-limit command to configure queue thresholds for a class, the WTD thresholds must be less than or equal to the queue-limit threshold configured with no qualifier. A queue size configured with no qualifier must be larger than any queue sizes configured with qualifiers.
- You cannot configure more than one unique threshold value for the WTD qualifiers (cos or DSCP) in the queue-limit command. However, there is no limit to the number of qualifiers that you can map to those thresholds. You can configure a second unique threshold value to set the maximum queue, using the queue-limit command with no qualifiers.
- In an output policy map, when you configure a queue-limit for a unique class, all other output policy maps must use the same format of qualifier type and qualifier value. Only queue-limit threshold values can be different. For example, when you configure class A queue-limit thresholds for DSCP 30 and DSCP 50 in policy-map1, and you configure class A queue-limits in policy-map 2, you must use DSCP 30 and DSCP 50 as qualifiers. You cannot use DSCP 20 and DSCP 40. The threshold values can be different, but different threshold values would create a new unique queue-limit configuration.

Beginning in privileged EXEC mode, follow these steps to use WTD to adjust the queue size for a traffic class:

Procedure

1. Use the **configure terminal** command to enter global configuration mode.

   Example

   ```
   Switch# configure terminal
   ```

2. Use the **policy-map** *policy-map-name* command to create a policy map and enter policy-map configuration mode.

   Example

   ```
   Switch(config)# policy-map policy-map-name
   ```

   For *policy-map-name*, specify the name of the policy map.

3. Use the **class** {*class-map-name* | **class-default**} command to select the default traffic class and enter policy-map class configuration mode.

   Example

   ```
   Switch(config-pmap)# class class-default
   ```

   The default class matches packets that the other traffic classes do not classify.

4. Use the **bandwidth** {*rate* | **percent** *value* | **remaining percent** *value*} command to configure a scheduling action for the traffic class.

Example

```
Switch(config-pmap-c)# bandwidth { rate | percent value | remaining percent
value }
```

Specify an absolute **rate**, a percentage of the total **bandwidth**, or a percentage of the **remaining** bandwidth.

5. Use the **queue-limit** [**cos** *value* | **DSCP** *value*] *number-of-packets* [**packets**] command to configure the queue size for the traffic class.

Example

```
Switch(config-pmap-c)# queue-limit [ cos value | dscp value ] number-of-packets
 [packets]
```

Specify the queue size for the traffic class.

- (Optional) Specify **cos value** to apply the queue limit to a CoS value. The range is from 0 to 7.
- (Optional) Specify **dscp** *dscp-value* to apply the queue limit to a DSCP value from 0 through 63.
- For **number-of-packets**, set the minimum threshold for WTD. The range is from 16 to 544, in multiples of 16, where each packet is a fixed unit of 256 bytes.

> **Note**
>
> For optimal performance, we strongly recommend that you configure the queue-limit to 272 or less.

The value is specified in packets by default, but the packets keyword is optional.

> **Note**
>
> Multiple output policy maps can use the same queue-limit configuration. However, these policy maps can have only three unique queue-limit configurations.

6. Use the **exit** command twice to return first to policy-map configuration mode and then to global configuration mode.

Example

```
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
```

7. Use the **interface** *interface-id* command to specify the interface to which you want to attach the policy map and enter interface configuration mode.

Example

```
Switch(config)# interface interface-id
```

8. Use the **service-policy output** *policy-map-name* command to attach the policy map created in Step 2 to the egress interface.

Example

```
Switch(config-if)# service-policy output policy-map-name
```

If the output policy map contains a fourth unique queue-limit configuration, the switch displays an error message and does not attach the policy map.

9. Use the **end** command to return to privileged EXEC mode.

Example

```
Switch(config-if)# end
```

Verification

Use the **show policy-map** [*policy-map-name* [**class** *class-map-name*]] command to verify your entries.

```
Switch# show policy-map [policy-map-name [class class-map-name]]
```

Optionally, use the **copy running-config startup-config** command to save your entries in the startup configuration file.

```
Switch# copy running-config startup-config
```

What to do next

After you have created an output policy map, you attach it to an egress port.

Use the **no** form of the appropriate command to delete an existing policy map or class map or to delete a WTD configuration.

This example shows a policy map with a specified bandwidth and queue size. Traffic that is not DSCP 30 is assigned a queue limit of 112 packets. Traffic with a DSCP value of 30 is assigned a queue-limit of 48 packets. All traffic not belonging to the class traffic is classified into class-default, which is configured with 10 percent of the total available bandwidth and a large queue size of 256 packets.

```
Switch(config)# policy-map gold-policy
Switch(config-pmap)# class traffic
Switch(config-pmap-c)# bandwidth percent 50
Switch(config-pmap-c)# queue-limit 112
Switch(config-pmap-c)# queue-limit dscp 30 48
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config-pmap)# class class-default
Switch(config-pmap-c)# bandwidth percent 10
Switch(config-pmap-c)# queue-limit 256
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface GigabitEthernet1/17
Switch(config-if)# service-policy output gold-policy
Switch(config-if)# exit
```

This example configures class A to match DSCP values and a policy map, PM1. The DSCP values of 30, 40, 50, and 60 are mapped to the maximum threshold of 112 packets.

```
Switch(config)# class-map match-any classA
Switch(config-cmap)# match ip dscp 30 40 50 60
```

```
Switch(config-cmap)# exit
Switch(config)# policy-map PM1
Switch(config-pmap)# class classA
Switch(config-pmap-c)# bandwidth percent 50
Switch(config-pmap-c)# queue-limit 112
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet 1/1
Switch(config-if)# service-policy output PM1
Switch(config-if)# exit
```

This example shows how to configure bandwidth and queue limit so that out-class1, out-class2, out-class3, and class-default get a minimum of 40, 20, 10 and 10 percent of the traffic bandwidth, respectively. The corresponding queue-sizes are set to 48, 32, 16 and 272 (256-byte) packets:

```
Switch(config)# policy-map out-policy
Switch(config-pmap)# class outclass1
Switch(config-pmap-c)# bandwidth percent 40
Switch(config-pmap-c)# queue-limit 48
Switch(config-pmap-c)# exit
Switch(config-pmap)# class outclass2
Switch(config-pmap-c)# bandwidth percent 20
Switch(config-pmap-c)# queue-limit 32
Switch(config-pmap-c)# exit
Switch(config-pmap)# class outclass3
Switch(config-pmap-c)# bandwidth percent 10
Switch(config-pmap-c)# queue-limit 16
Switch(config-pmap-c)# exit
Switch(config-pmap)# class class-default
Switch(config-pmap-c)# bandwidth percent 10
Switch(config-pmap-c)# queue-limit 272
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet 1/1
Switch(config-if)# service-policy output out-policy
Switch(config-if)# exit
```

Display QoS information

Lists commands for displaying QoS information and related configuration details.

To display QoS information, use one or more of the privileged EXEC commands. You can also test full-path QoS in both directions on an interface by configuring Ethernet terminal loopback using the ethernet loopback facility interface configuration command. In terminal loopback mode, the port appears to be up but the link is actually down and no packets are sent out. Configuration changes on the port immediately affect the traffic being looped back.

This section contains these topics:

- QoS statistics
- Restrictions and limitations

Table 3: Commands for displaying standard QoS information

Command	Purpose
show class-map [*class-map-name*]	Displays QoS class-map information for all class maps or the specified class map.

Command	Purpose
show policer aggregate [*aggregate-policer-name*]	Displays information about all aggregate policers or the specified aggregate policer.
show policy-map [**policy-map-name** | **interface** [*interface-id*] [**input** | **output**] [*class class-name*]]	Displays QoS policy map information for the specified policy map name, interface, input or output policy maps, or policy-map class.
show running-config	Displays the configured class maps, policy maps, table maps, and aggregate policers.

QoS statistics

Provides methods to display QoS input and output policy-map statistics for monitoring network performance and traffic management.

There are several ways to display QoS input and output policy-map statistics.

Input policy map statistics

For input policy maps, you can use the **show policy-map interface** [*interface-id*] privileged EXEC command to display per-class per-policer conform and exceed statistics. Policer conform statistics are the number of packets that conform to the configured policer profile; policer exceed statistics are the number of packets that exceed the configured policer profile. The switch does not support per-class classification statistics, but you can determine these statistics by configuring policing at line rate for the class. In this case, no packets exceed the configured policer profile, and the policer conform statistics would equal the class classification statistics.

Output policy map statistics

For output policy maps, you can use the **show policy-map interface** [*interface-id*] command to display per-class classification statistics that show the total number of packets that match the specified class. This count includes the total number of packets that are sent and dropped for that class. You can use the same command to view the per-class tail drop statistics.

Queue and packet processing details

Use the **show platform hardware qos asic 0 port** [*port id*] command to view details about queue and packet processing.

Use the **show platform pm port-map** command to obtain the Port ID:

```
Switch# show platform pm port-map
interface gid gpn asic slot unit gpn-idb
----------------------------------------
Te1/1 1 1 0/25 1 1 Yes
Te1/2 2 2 0/27 1 2 Yes
Gi1/3 3 3 0/2 1 3 Yes
Gi1/4 4 4 0/3 1 4 Yes
Gi1/5 5 5 0/0 1 5 Yes
Gi1/6 6 6 0/1 1 6 Yes
Gi1/7 7 7 0/6 1 7 Yes
Gi1/8 8 8 0/7 1 8 Yes
Gi1/9 9 9 0/4 1 9 Yes
Gi1/10 10 10 0/5 1 10 Yes
Gi2/1 11 11 0/10 2 1 Yes
Gi2/2 12 12 0/11 2 2 Yes
Gi2/3 13 13 0/8 2 3 Yes
Gi2/4 14 14 0/9 2 4 Yes
```

```
Gi2/5 15 15 0/14 2 5 Yes
Gi2/6 16 16 0/15 2 6 Yes
Gi2/7 17 17 0/12 2 7 Yes
Gi2/8 18 18 0/13 2 8 Yes
Gi2/9 19 19 0/18 2 9 Yes
Gi2/10 20 20 0/19 2 10 Yes
Gi2/11 21 21 0/16 2 11 Yes
Gi2/12 22 22 0/17 2 12 Yes
Gi2/13 23 23 0/22 2 13 Yes
Gi2/14 24 24 0/23 2 14 Yes
Gi2/15 25 25 0/20 2 15 Yes
Gi2/16 26 26 0/21 2 16 Yes
Switch#
```

```
Switch# show platform hardware qos asic 0 port 1
Dumping QoS settings for port 1Port | Trust | Modify | Modify | Default |
Default| Mode | DSCP | UP | QosProfile | UP---- |------ | ------ | ------ |
---------- | -------1 | L2+L3 | No | No | 65 | 0Queue[0]: Thresh[0]:0 Drops[0]:0
 Thresh[1]:0 Drops[1]:0 Thresh[2]:22 Drops[2]:0Queue[1]: Thresh[0]:0 Drops[0]:0
 Thresh[1]:0 Drops[1]:0 Thresh[2]:0 Drops[2]:0Queue[2]: Thresh[0]:0 Drops[0]:0
 Thresh[1]:0 Drops[1]:0 Thresh[2]:0 Drops[2]:0Queue[3]: Thresh[0]:0 Drops[0]:0
 Thresh[1]:0 Drops[1]:0 Thresh[2]:0 Drops[2]:0Queue[4]: Thresh[0]:0 Drops[0]:0
 Thresh[1]:0 Drops[1]:0 Thresh[2]:0 Drops[2]:0Queue[5]: Thresh[0]:0 Drops[0]:0
 Thresh[1]:0 Drops[1]:0 Thresh[2]:68 Drops[2]:99932Queue[6]: Thresh[0]:0
Drops[0]:0 Thresh[1]:0 Drops[1]:0 Thresh[2]:196 Drops[2]:99932Queue[7]:
Thresh[0]:0 Drops[0]:0 Thresh[1]:0 Drops[1]:0 Thresh[2]:104455 Drops[2]:0

Dumping Ingress QoS stats 0

class [0]:    in_class_policer:
    redCntr_packets 0
    yellowCntr_packets 0
    greenCntr_packets 0

class [1]:    in_class_policer:
    redCntr_packets 0
    yellowCntr_packets 0
    greenCntr_packets 0

class [2]:    in_class_policer:
    redCntr_packets 0
    yellowCntr_packets 0
    greenCntr_packets 0

class [3]:    in_class_policer:
    redCntr_packets 0
    yellowCntr_packets 0
    greenCntr_packets 0

class [4]:    in_class_policer:
    redCntr_packets 0
    yellowCntr_packets 0
    greenCntr_packets 0

class [5]:    in_class_policer:
    redCntr_packets 0
    yellowCntr_packets 0
    greenCntr_packets 0

class [6]:    in_class_policer:
    redCntr_packets 0
    yellowCntr_packets 0
```

```
    greenCntr_packets 0

class [7]:    in_class_policer:
    redCntr_packets 0
    yellowCntr_packets 0
    greenCntr_packets 0
Switch#
```