CHAPTER 33

Configuring Quality of Service (QoS)

This chapter describes how to configure quality of service (QoS) on the ME 3800X and ME 3600X switches by using the modular QoS command-line interface (MQC) commands. With QoS, you can provide preferential treatment to certain types of traffic at the expense of other types. When you do not configure QoS, the switch offers best-effort service to each packet, regardless of the packet contents or size. MQC provides a hierarchical configuration framework for prioritizing or limiting specific streams of traffic.

QoS includes traffic classification, marking, policing, queuing, and scheduling configured with service policies that are attached to ingress and egress targets. On the ME 3800X and ME 3600X switches, targets can be switchports or Ethernet Flow Points (EFPs), also referred to as service instances. The switches do not support the service policies attached to the EtherChannel port channels although you can attach them to ports that belong to an EtherChannel as long as there are no EFPs configured on the EtherChannel.

Ingress QoS includes classification, marking, and policing. Classification can be based on the class of service (CoS), Differentiated Services Code Point (DSCP), IP precedence, or the multiprotocol label switching (MPLS) experimental (EXP) value in the inbound packet. You can classify based on Layer 2 MAC, IP-standard, or IP-extended access control lists (ACLs).

Egress QoS supports the same classifications as ingress QoS except for ACLs, and also includes classification based on QoS group or discard class. Egress QoS also includes queueing based on the weighted tail drop (WTD) algorithm, scheduling based on shaped weights, and an egress priority queue.

You can also use hierarchical QoS to classify, police, mark, queue, and schedule inbound or outbound traffic. You can define a policy map for the first, second, or third level of the hierarchy. Hierarchical QoS offers classification based on the CoS, DSCP, IP precedence, or the MPLS EXP bits in the packet, and classifying a packet based on its VLAN. The switch supports two-rate, three-color policing at different levels. Drop policy actions include passing the packet through without modification is supported and ingress and egress however, marking down the CoS, DSCP, IP precedence, or the MPLS EXP bits in the packet; or dropping the packet is only supported at ingress.

For more information about Cisco IOS MQC commands, see the “Cisco IOS Quality of Service Solutions Command Reference:”

http://www.cisco.com/en/US/docs/ios/qos/command/reference/qos_book.html

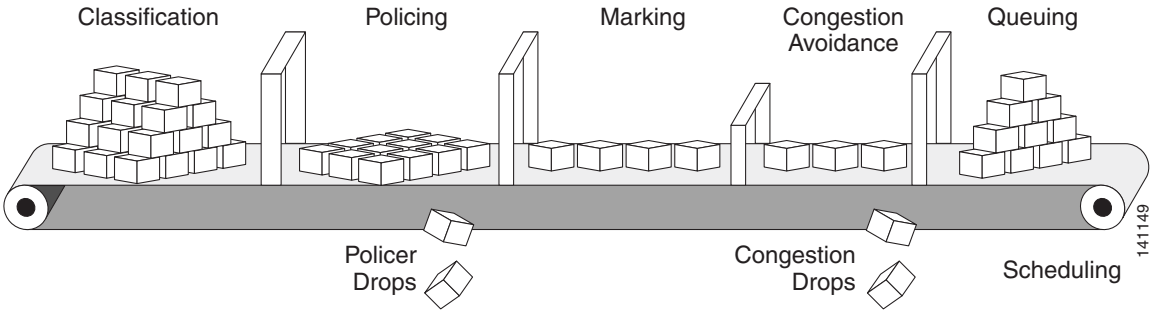
For complete syntax and usage information for the platform-specific commands used in this chapter, see the command reference for this release.

Understanding QoS

When networks operate on a best-effort delivery basis, all traffic has equal priority and an equal chance of being delivered in a timely manner. When congestion occurs, all traffic has an equal chance of being dropped. When you configure QoS, you can select specific network traffic, prioritize it according to its relative importance, and use traffic-management techniques to provide preferential treatment. Implementing QoS in your network makes network performance more predictable and bandwidth utilization more effective.

Figure 33-1 shows the MQC model.

Figure 33-1 Modular QoS CLI Model



Basic QoS includes these actions.

- Packet classification organizes traffic on the basis of whether or not the traffic matches a specific criteria. When a packet is received, the switch identifies all key packet fields: CoS, DSCP, IP precedence, or MPLS EXP. The switch classifies the packet based on this content or based on an ACL lookup. 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. See the "Classification" section on page 33-5.
- Packet policing determines whether a packet is in or out of profile by comparing the rate of the incoming traffic to the configured policer. You can configure a committed information rate (CIR) and peak information rate (PIR) and set actions to perform on packets that conform to the 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). See the "Policing" section on page 33-16.
- All packets that belong to a classification can be remarked. When you configure a policer, packets that meet or exceed the permitted bandwidth requirements (bits per second) can be conditionally passed through, dropped, or marked. You use the **police** command to conditionally mark incoming packets based on the CIR and PIR. You use the **set** command to unconditionally mark packets. See the "Packet Marking" section on page 33-19.
- Congestion management uses queuing and scheduling algorithms to queue and sort traffic that is leaving a port. The switch supports these scheduling and traffic-limiting features: class-based weighted fair queuing (CBWFQ), class-based traffic shaping, port shaping, and class-based priority queuing. You can provide guaranteed bandwidth to a particular class of traffic while still servicing other traffic queues. See the "Congestion Management and Scheduling" section on page 33-24.
- Queuing on the switch uses the WTD algorithm, a congestion-avoidance mechanism. WTD differentiates traffic classes and regulates the queue size based on the classification. See the "Congestion Avoidance and Queuing" section on page 33-20.

This section includes information about these topics:

Modular QoS CLI Configuration

With the modular QoS CLI, you create traffic policies and attach these policies to physical interfaces or EFP service instances. A traffic policy contains a traffic class and one or more QoS features. You define a traffic class to classify traffic, use a traffic policy to define how to treat the classified traffic, and attach the policy to a port or service instance to create a service policy.

Step 1 Define a traffic class.

Use the **class-map** global configuration command to define a traffic flow or class and to enter class-map configuration mode. A traffic class contains:

- A name—You name the traffic class in the **class-map** command line to enter class-map configuration mode.
- (Optional) Keywords to evaluate the match commands, either **class-map match-any** or **class-map match-all**. By default, match-all is supported with a class map is defined and match-any is not specified. Only one match statement is allowed for match-all, except for outer VLAN and inner VLAN, or outer CoS and inner CoS matches for 802.1Q tunneling (QinQ) packets.
- A series of **match** class-map configuration commands to specify criteria for classifying packets. Criteria can include matching an access group defined by an ACL or matching a specific list of COS, DSCP, IP precedence, or MPLS EXP values. 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 do not meet any of the matching criteria are classified as members of the default traffic class.

Note For exceptions to the list of match statements, see the "Classification" section on page 33-5.

Step 2 Associate policies and actions with each traffic class.

Use the **policy-map** global configuration command to create a traffic policy and to enter policy-map configuration mode. A traffic policy specifies the traffic class to act on and defines the QoS features to associate with the specified traffic class. A traffic policy contains a name, a traffic class (specified with the **class** policy-map configuration command), and the QoS policies configured in the class.

- A name—You name the traffic policy in the **policy-map** command line to enter policy-map configuration mode.
- A traffic class—Use the **class** policy-map configuration command to enter the name of the traffic class used to classify traffic to the specified policy, and enter policy-map class configuration mode.

- The QoS features to apply to the classified traffic. These include the **set** or **police** 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.

Step 3 Attach the traffic policy to a target, which can be an interface or an EFP service instance.

Use the **service-policy** interface configuration command to attach the policy map to a target and to specify if the policy should be applied to packets that enter or leave the target. For example, entering the **service-policy output policy1** interface configuration command attaches all the characteristics of the traffic policy named *class1* to the interface. Entering the **service-policy output policy1** service instance configuration command attaches all the characteristics of the traffic policy named *class1* to the EFP service policy. All packets leaving the target are evaluated according to the criteria specified in the traffic policy *class1*.

Note If you enter the **no policy-map** *policy-map-name* global configuration command to delete a policy map that is attached to an interface or service instance, 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 GigabitEthernet0/1
```

Hierarchical QoS

Hierarchical QoS configuration involves traffic classification, policing, queuing, and scheduling. You can create a hierarchy by associating a class-level policy map with a VLAN-level policy map, by associating that VLAN-level policy map with a physical-level policy map, and by attaching the physical-level policy map to a port or EFP. You can omit hierarchical levels, but the order of levels (class level, VLAN level, and then physical level) must be preserved.

You can configure three QoS levels in the hierarchy:

- Class level—You configure this level of the hierarchy by matching CoS, DSCP, IP precedence, MAC ACLs, IP ACLs, QoS groups, discard-class, or MPLS EXP bits in the packet by using the **match** {**access-group** *name* | **cos** [**inner**] *cos-list* | **discard-class** *value* | **dscp** *dscp-list* | **ip precedence** *ip-precedence-list* | **mpls experimental** *exp-list*} | **qos-group** *value* class-map configuration command.

  At the class level, you can use policy-map class configuration commands to:

  - Configure policer drops by using the **police cir** or **police cir percent** command.
  - Configure tail drop policies by using the **queue-limit** command.
  - Modify the traffic class by setting Layer 2 and Layer 3 QoS fields by using the **set** commands.
  - Configure scheduling by using the **bandwidth** or the **priority** command.
  - Configure traffic shaping by using the **shape** command.
- VLAN level—You configure per-VLAN QoS by entering the match vlan *vlan-id* or **match vlan-inner** *vlan-id* class-map configuration command for one or more VLANs.

At the VLAN level, you can:

- Configure the VLANs to police ingress traffic by using the **police cir** or **police cir percent policy** command.
- Configure unconditional marking on ingress traffic by using the **set** command.
- Configure the queue to share the available port bandwidth and enable CBWFQ by using the **bandwidth** command.
- Configure traffic shaping by using the **shape** command.

You can also associate a previously defined child policy at the class level with a new service policy by using the **service-policy** policy-map class configuration command to apply the class-level policy only to traffic that matches the VLAN class. You cannot mix VLAN-level and class-level matches within a class map.

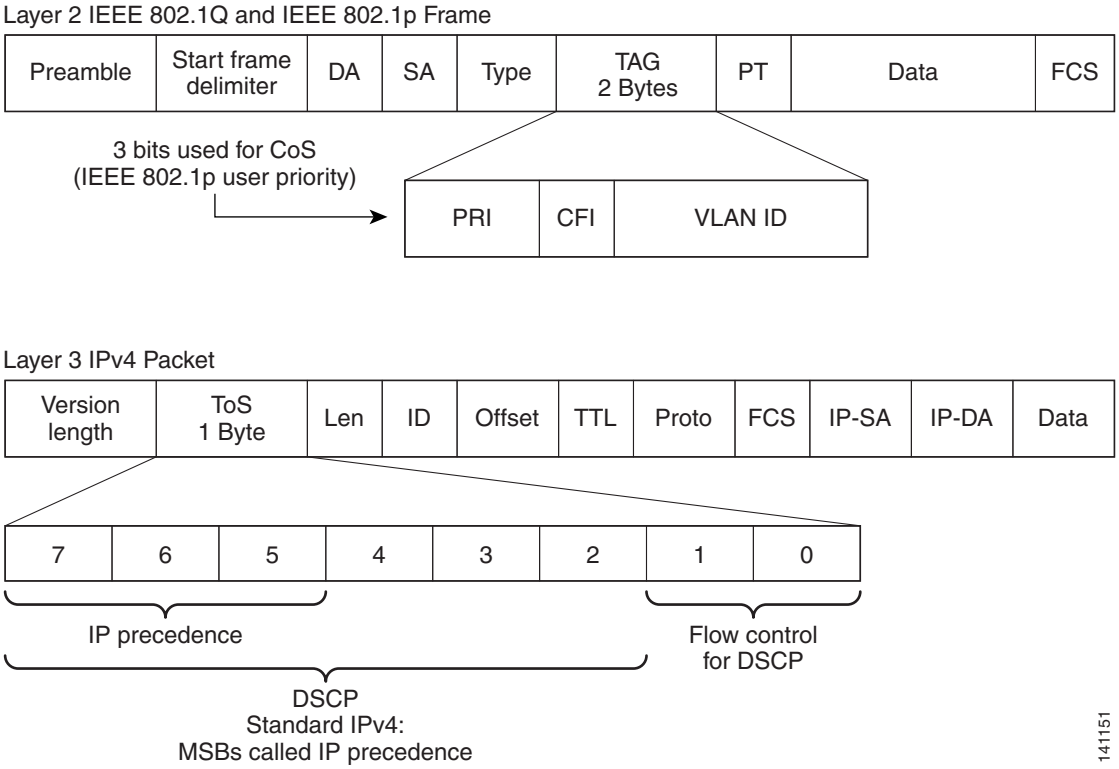
- Physical level—You can shape or police only the class-default class at the physical level of the hierarchy by using the **shape**, **police cir**, or **police cir percent** policy-map class configuration command. Within a policy map, the **class-default** applies to all traffic that is not explicitly matched within the policy map but that does match the parent policy. If no parent policy is configured, the parent policy represents the physical port.
  - Configure unconditional marking by using the **set** command.
  - In a physical-level policy map, **class-default** is the only class that you can configure. You use the **service-policy** {**input** | **output**} policy-map-name interface configuration command to attach a hierarchical policy to a physical port or to an EFP.

Classification

Classification distinguishes one kind of traffic from another by examining the fields in the packet header. When a packet is received, the switch examines the header and identifies all key packet fields. A packet can be classified based on an ACL, on the DSCP, the CoS, IP precedence, or MPLS EXP value in the packet, or by the VLAN ID. Figure 33-2 has examples of classification information carried in a Layer 2 or a Layer 3 IP packet header, using six bits from the deprecated IP type of service (ToS) field to carry the classification information.

- Layer 2 frame headers have a 2-byte Tag Control Information field that carries the CoS value, called the User Priority bits, in the 3 most-significant bits, and the VLAN ID value in the 12 least-significant bits. Layer 2 CoS values range from 0 to 7.
- Layer 3 IP packets can carry either an IP precedence value or a DSCP value. QoS supports the use of either value because DSCP values are backward-compatible with IP precedence values. IP precedence values range from 0 to 7. DSCP values range from 0 to 63. MPLS EXP values range from 0 to 7.

Figure 33-2 QoS Classification Layers in Frames and Packets



The match Command

In class-map configuration mode, you use the **match** class-map configuration command to define the match criterion for the traffic. You can also create a class map that requires that all matching criteria in the class map be in the packet header by using the **class map match-all** *class-map name* global configuration command.

Note The **match-all** keyword is supported only for outer and inner VLAN, or outer and inner CoS matches for QinQ packets and is rejected for all other mutually exclusive match criteria. You can configure only one match entry in a **match-all** class map

You can use the **class map match-any** *class-map name* global configuration command to define a classification with any of the listed criteria.

In class-map configuration mode, you use the **match** command to specify the classification criteria. If a packet matches the configured criteria, it belongs to a specific class and is forwarded according to the specified policy. For example, you can use the **match** class-map command with CoS, IP DSCP, IP precedence, or MPLS EXP values.You can also match an access group, a QoS group, or a VLAN ID or inner VLAN ID or VLAN ID range for per-port, per-VLAN QoS.

For an input policy map, you cannot configure both an IP classification (match ip dscp, **match ip precedence**, **match ip acl**) and a non-IP classification (**match mac acl**) in the same policy map or class map.

This example shows how to create a class map *example* to define a class that matches any of the listed criteria. In this example, if a packet is received with the DSCP equal to 32 or a 40, the packet is identified (classified) by the class map.

```
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
```

VLAN Match Support

The VLAN Match support feature allows classification based on VLAN on the main interface, which has EVC service instances configured on that interface.

Support VLAN-based policy on an EVC physical-port with the following EFP configuration:

- EFP VLAN encapsulation = Bridge-domain ID
- EFP rewrite = pop-1 ingress symmetric
- VLAN match in the class-map must be same as the Bridge-domain ID

Restrictions and Guidelines

The following restrictions apply to VLAN match support:

- The feature is only supported on main interfaces where the EVC Bridge Domain is configured.
- VLAN classification based policy-map, applied on the main interface where QoS on EVC is also configured, is not supported.
- We recommend you not use this feature with huge scale EVC configuration on the main interface as you may run out TCAMS.
- The VLAN match feature is supported on egress only.

The following example shows classification based on VLAN on the main interface, where an EVC is configured.

```
Switch(config)# class-map match-any vlan_class
Switch(config-cmap)# match vlan 200
Switch(config-cmap)# policy-map main-interface-policy
Switch(config-pmap)# class vlan_class
```

```
Switch(config-pmap-c)# exit
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# service-policy output main-interface-policy
Switch(config-if)# service instance 1 Ethernet
Switch(config-if-srv)# encapsulation dot1q 200
Switch(config-if-srv)# rewrite ingress tag pop 1 symmetric
Switch(config-if-srv)# bridge-domain 200
Switch(config-if-srv)# end
```

Classification Based on Layer 2 CoS

You use the **match cos** command to classify Layer 2 traffic based on the CoS value, which ranges from 0 to 7.

This example shows how to create a class map to match a CoS value of 5:

```
Switch(config)# class-map premium
Switch(config-cmap)# match cos 5
Switch(config-cmap)# exit
```

Classification Based on IP Precedence

You can classify IPv4 traffic based on the packet IP precedence values, which range from 0 to 7. This example shows how to create a class map to match an IP precedence value of 4:

```
Switch(config)# class-map sample
Switch(config-cmap)# match ip precedence 4
Switch(config-cmap)# exit
```

Classification Based on IP DSCP

When you classify IPv4 traffic based on the IP DSCP value and enter the **match ip dscp** class-map configuration command, 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* delivers IP packets in four different AF classes: AF11-13 (the highest), AF21-23, AF31-33, and AF41-43 (the lowest). 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 class provides the highest 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).

CoS Mapping

The switch uses EVC and EFPs to support VLAN mapping from the customer VLAN-ID (C-VLAN) to a service-provider VLAN-ID (S-VLAN). See the "Configuring IEEE 802.1Q Tunneling and Layer 2 Protocol Tunneling Using EFPs" section on page 12-19.

For QoS, you can set the service-provider CoS (S-CoS) from either the customer CoS (C-CoS) or the customer DSCP (C-DSCP) value. You can map the inner CoS to the outer CoS for any traffic with traditional 802.1Q tunneling (QinQ). This allows copying the customer CoS into the service provider network.

By default, the switch supports C-CoS to S-CoS propagation for QinQ. When you configure QinQ, you can also set the S-CoS from C-DSCP.

Configuring CoS matching on EFPs configured for tunneling:

- On service instances configured for 802.1Q tunneling, the CoS value of the VLAN tag (inner VLAN or C-VLAN) received on the interface (C-CoS) is automatically reflected in the tunnel VLAN tag (outer VLAN or S-VLAN) by default.
- The **set cos** policy-map class configuration commands always apply to the outer-most VLAN tag after processing is complete, that is the S-VLAN-ID. For example, in 802.1Q tunnels, entering a **set cos** command changes only the CoS value of the outer tag of the encapsulated packet.

Note Although you configure the command at input, because the switch supports only egress push, this affects *only* the CoS value of the tag imposed on egress.

- When you configure a policy by entering the **match dscp** class map configuration command and you enter the **set cos**-policy-map class configuration command for QinQ EFPs, a DSCP match sets the outer CoS of the encapsulated value.

Note As in the previous case, the command configured at input affects *only* the CoS value of the tag imposed at egress.

- You can set DSCP based on matching the outer VLAN.
- If you enter the **match cos** command on EFPs configured for QinQ, the match is to the incoming CoS (C-CoS).

The same CoS mapping rules also apply to EFP rewrite operations (see the "Rewrite Operations" section on page 12-7) when you use the **rewrite ingress tag pop symmetric** service instance command for VLAN translation.

You can also configure outgoing CoS on an 802.1Q trunk port to simulate CoS mapping.

Ingress Classification Based on QoS ACLs

You can use IP standard, IP extended, or Layer 2 MAC ACLs to define a group of packets with the same characteristics (*class*). In the QoS context, the permit and deny actions in the access control entries (ACEs) have different meanings than do security ACLs. QoS policies do not match ACLs that use the **deny** keyword.

- If a match with a permit action is encountered (first-match principle), the specified QoS-related action is taken.
- If a match with a deny action is encountered, the ACL being processed is omitted, and the next ACL is processed.
- If no match with a permit action is encountered and all the ACEs have been examined, no QoS processing occurs on the packet, and the switch offers best-effort service to the packet.
- If multiple ACLs are configured on an interface, the lookup stops after the packet matches the first ACL with a permit action, and QoS processing begins.

Note When you create an access list, remember that the end of the access list contains an implicit deny statement for everything if it did not find a match before reaching the list end.

You implement IP ACLs to classify IP traffic by using the **access-list** global configuration command. You implement Layer 2 MAC ACLs to classify non-IP traffic by using the **mac access-list extended** global configuration command. The switch supports MAC ACLs only with destination addresses.

Not all IP ACL options are supported in QoS ACLs. Only these protocols are supported for **permit** actions in an IP ACL: ICMP, IGMP, GRE, IPINIP, TCP, and UDP. Within a protocol, for IP source and destination, the switch supports only the source or destination IP address, host, or any. For matching criteria, the switch supports only DSCP, time-range, and ToS. See the "Using ACLs to Classify Traffic" section on page 33-36 for more specific information. When you define a class map with the ACL, you can add the class to a policy.

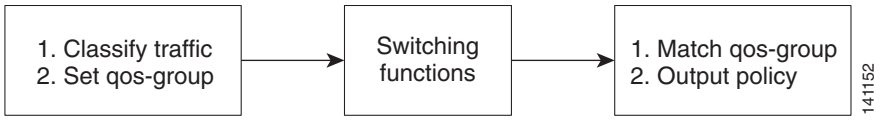
You can attach a policy that includes unsupported QoS IP ACL options to the target, but QoS ignores the unsupported options. If you modify an IP ACL in a policy map that is already attached to a target and the modification causes the policy to become invalid, the policy is detached from the target.

Classification Based on QoS Groups

A QoS group is an internal label used by the switch to identify packets as a members of a specific class. The label is not part of the packet header and is restricted to the switch that sets the label and not communicated between devices. QoS groups provide a way to tag a packet for subsequent QoS action without explicitly marking (changing) the packet.

A QoS group is identified at ingress and used at egress. It is assigned in an input policy to identify packets in an output policy. See Figure 33-3.

Figure 33-3 QoS Groups



You use QoS groups to aggregate different classes of input traffic for a specific action in an output policy. For example, you can classify an ACL on ingress by using the set qos-group command and then use the match qos-group command in an output policy.

```
Switch(config)# class-map acl
Switch(config-cmap)# match access-group name acl
Switch(config-cmap)# exit
```

Input policy map:

```
Switch(config)# policy-map set-qos-group
Switch(config-pmap)# class acl
Switch(config-pmap-c)# set qos-group 5
Switch(config-cmap-c)# exit
```

Output policy map:

```
Switch(config)# policy-map shape
Switch(config-pmap)# class qos-group 5
Switch(config-pmap-c)# shape average 10m
Switch(config-cmap-c)# exit
```

You can use QoS groups to aggregate multiple input streams across input classes and policy maps to have the same QoS treatment on the egress port. Assign the same QoS group number in the input policy map to all streams that require the same egress treatment, and match the QoS group number in the output policy map to specify the required queuing and scheduling actions.

You can also use QoS groups to implement MPLS tunnel mode. In this mode, the output per-hop behavior of a packet is determined by the input EXP bits, but the packet remains unmodified. You match the EXP bits on input, set a QoS group, and then match that QoS group on output to obtain the required QoS behavior.

To communicate an ACL classification to an output policy, you assign a QoS number to specify packets at ingress. This example identifies specific packets as part of QoS group 1 for later processing in an output policy:

```
Switch(config)# policy-map in-gold-policy
Switch(config-pmap)# class in-class1
Switch(config-pmap-c)# set qos-group 1
Switch(config-cmap-c)# exit
Switch(config-cmap)# exit
```

You use the **set qos-group** command only in an input policy. The assigned QoS group identification is then used in an output policy with no mark or change to the packet. You use the **match qos-group** in the output policy. You cannot configure **match qos-group** for an input policy map.

This example creates an output policy to match the QoS group created in the input policy map *in-gold-policy*. Traffic internally tagged as *qos-group 1* is identified and processed by the output policy.

```
Switch(config)# class-map out-class1
Switch(config-cmap)# match qos-group 1
Switch(config-cmap)# exit
```

The switch supports a maximum of 100 QoS groups.

Classification Based on Discard Class

A discard class is very similar to a QoS group in that it is a virtual packet marking that is carried in the packet within a single device. The difference is that a QoS group defines the complete QoS behavior for a packet, while a discard class only communicates the drop precedence of the packet during congestion management. For example, a packet could be classified on input into one QoS group, but within that QoS group, a policer could mark one of three discard classes, depending on whether the packet was determined to conform to, exceed, or violate the configured specifications. On output, a class would match the QoS group. but you could configure three different drop curves, one for each of the discard classes. The discard class value ranges from 0 to 7.

Classification Based on VLAN IDs

With classification based on VLAN IDs, you can apply QoS policies to frames carried on a user-specified VLAN for a given interface. You can use hierarchical policy maps for per-VLAN classification on trunk ports. Per-VLAN classification is not required on access ports because access ports carry traffic for a single VLAN.

You use the **match vlan** *vlan-id* class-map configuration command to classify based on the outer VLAN. Use the **match vlan inner** *vlan-id* class-map configuration command to classify based on the inner VLAN

QoS-Context Manager

QoS-Contest Manager allocates QoS-context for s-vlan and c-vlan matches. To display qos-context manager information use the **show platform qos context struct** *bridge domain* command.

QOS-contexts are allocated only for intersecting combinations of a configured class-map vlans and the vlans present in encapsulation

Packets can come from

- Multiple ingress rewrite encapsulation types (pop 0, pop 1 and pop 2)
- With and without Ingress Service policies

For every egress vlan combination, qos-context is allocated in the ingress tcam based on the ingress rewrite type

Restrictions and limitations

QoS-context manager has the following limitations:

- There are 63 qos-contexts available per bridge-domain.
- QoS-Context manager is fixed and cannot be changed

QoS-context depends on the following:

- Presence of ingress service policy.
- Presence of ingress rewrite type (pop 0, pop 1, pop 2)#
- Bridge-domain or xconnect must be configured on the service instance
- Applies to both ME3600X and ME 3800X

Note Where ingress service policy is not configured the ingress rewrite type is used to allocate qos-contexts.

Working Configuration

The following is an example of a working configuration for allocation of qos-context:

Policy Used:

```
class-map match-any ME-CUSTOMERS-CLASS
match vlan 1500-3299
class-map match-any VOIP-CLASS
match vlan 1201 1301-1302
class-map match-any MGMT-CLASS
match vlan 1050-1099
class-map match-any INTERNET-CLASS
match vlan 1101
bandwidth remaining percent 10
policy-map LLU-NAME-PER-SERVICE-POLICY
class VOIP-CLASS
Priority
class MGMT-CLASS
bandwidth remaining percent 15
class INTERNET-CLASS
bandwidth remaining percent 75
class class-default
bandwidth remaining percent 10
```

Configuration:

```
Switch# sh run int gi0/1
Building configuration...
Current configuration : 336 bytes
!
interface GigabitEthernet0/1
port-type nni
 switchport trunk allowed vlan none
switchport mode trunk
mtu 9216
service instance 100 ethernet
  encapsulation dot1q 1-1080
  service-policy output LLU-NAME-PER-SERVICE-POLICY
  bridge-domain 3600
!
End
```

Based on the class-maps in the applied policy, 31 vlans in the class-maps intersecting with the encap vlans "encapsulation dot1q 1-1080" (match vlan 1050-1099)

2 Qos-contexts are allocated for every vlan match in the egress side of this evc, total qos-contexts 31*2=62

Policy is accepted as 63 qos-contexts are supported per bridge-domain

Failed Configuration:

The following is an example of a failed configuration for allocation of qos-contexts:

```
Switch# confure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# interface GigabitEthernet0/1
Switch(config-if)# service instance 100 ethernet
Switch(config-if-srv)# encapsulation dot1q 1-1081
QoS: Maximum Egress QosContexts consumed in Bridge-Domain: 3600
QoS: Detaching output service-policy LLU-NAME-PER-SERVICE-POLICY from efp 100
Switch(config-if-srv)#
*Apr 21 14:40:37.443: %QOSMGR-3-EQOS_CXT_EXCEEDED: Maximum Egress QosContexts consumed in
the Bridge-Domain
```

Based on the class-maps in the applied policy, 32 vlans in the class-maps intersecting with the encap vlans "encapsulation dot1q 1-1081" (match vlan 1050-1099)

2 Qos-contexts are allocated for every vlan match in the egress side of this evc, total qos-contexts 32*2=64

Policy is rejected as only 63 qos-contexts are supported per bridge-domain

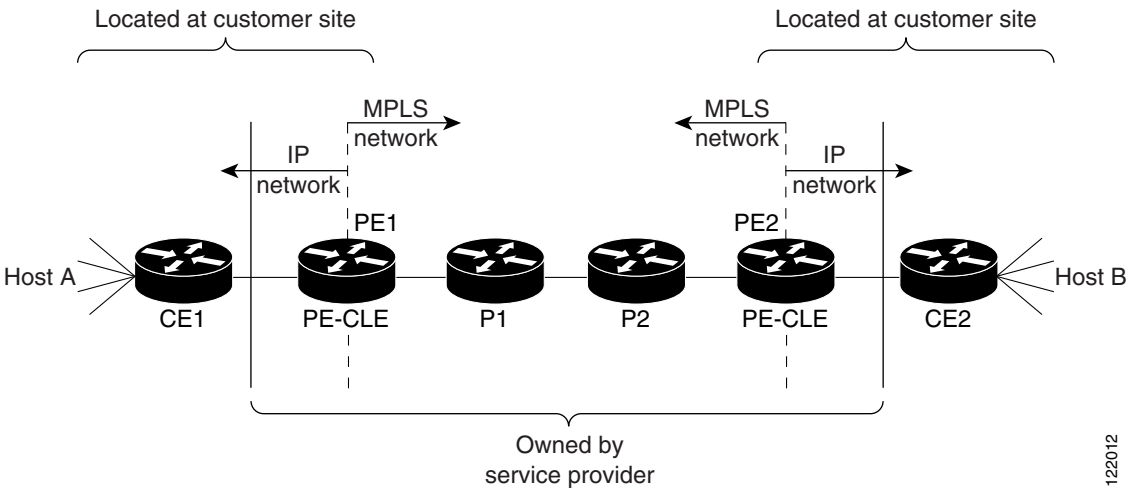
Classification for MPLS and EoMPLS

In an MPLS network, QoS can be specified in different ways. For example, the IP precedence field (the first 3 bits of the DSCP field in the header of an IP packet) can specify the QoS value to give the packet. If the network is an MPLS network, the IP precedence bits are copied into the MPLS EXP field at the edge of the network. If a service provider wants to set a QoS value for an MPLS packet to a different value, instead of overwriting the value in the IP precedence field that belongs to a customer, the service provider can set the MPLS experimental field. The IP header remains available for the customer's use, and the QoS of an IP packet is not changed as the packet travels through the MPLS network.

By choosing different values for the MPLS experimental field, you can mark packets based on their characteristics, such as rate or type, so that packets have the priority that they require during periods of congestion.

Figure 33-4 shows an MPLS network that connects two sites of an IP network that belongs to a customer.

Figure 33-4 MPLS Network Connecting Two Customer Sites



PE1 and PE2 are customer-located routers at the boundaries between the MPLS network and the IP network and are the ingress and egress provider-edge devices. CE1 and CE2 are customer edge devices. P1 and P2 are service provider routers within the core of the service-provider network.

Packets arrive as IP packets at PE1, the ingress provider-edge router, and PE1 sends the packets to the MPLS network as MPLS packets. Within the service-provider network, there is no IP precedence field for the queuing mechanism to look at because the packets are MPLS packets. The packets remain as MPLS packets until they arrive at PE2, the egress provider-edge router. PE2 removes the label from each packet and forwards the packets as IP packets.

Service providers can use MPLS QoS to classify packets according to the type, and other factors by setting (marking) each packet within the MPLS experimental field without changing the IP precedence or DSCP field. You can use the IP precedence or DSCP bits to specify the QoS for an IP packet and use the MPLS experimental bits to specify the QoS for an MPLS packet. In an MPLS network, configure the MPLS experimental field value at PE1 (the ingress router) to set the QoS value in the MPLS packet.

It is important to assign the correct priority to a packet. The packet priority affects how the packet is treated during periods of congestion. For example, service providers have service-level agreements with customers that specify how much traffic the service provider has agreed to deliver. To comply with the agreement, the customer must not send more than the agreed-upon rate. Packets are considered to be in-rate or out-of-rate. If there is congestion in the network, out-of-rate packets might be dropped more aggressively.

MPLS QoS matches only valid MPLS packets. On input, the match is performed before any label processing on the packet. On output, the match is performed on the final packet after all label operations are performed. See the "Configuring MPLS and EoMPLS QoS" section on page 33-72.

EoMPLS and QoS

EoMPLS supports QoS by using three experimental bits in a label to determine the priority of packets. To support QoS between label edge routers (LERs), you set the experimental bits in both the virtual connection and the tunnel labels. EoMPLS QoS classification occurs on ingress, and you can match on Layer 3 parameters (such as IP or DSCP), and Layer 2 parameters (CoS). Mapping does not occur by default and you must set MPLS experimental bits at ingress. See the "Configuring MPLS and EoMPLS QoS" section on page 33-72 for more information about EoMPLS and QoS.

Table Maps

You can use table maps to manage a large number of traffic flows with a single command. You can specify table maps in **set** commands and use them as mark-down mapping for the policers. You can also use table maps to map an incoming QoS marking to a replacement marking without having to configure a large number of explicit matches and sets. Table maps are used only in input policy maps.

Table maps are not supported under class-default.

Table maps can be used to:

- Correlate specific CoS, DSCP, or IP precedence values to specific CoS, DSCP, or IP precedence values
- Mark down a CoS, DSCP, or IP precedence value
- Assign defaults for unmapped values

A table map includes one of these default actions:

- **default** *default-value*—applies a specific default value (0 to 63) for all unmapped values
- **default copy**—maps all unmapped values to the equivalent value in another qualifier
- **default ignore**—makes no changes for unmapped values

This example creates a table to map specific CoS values to DSCP values. The **default** command maps all unmapped CoS values to a DSCP value of 63.

```
Switch(config)# table-map cos-dscp-tablemap
Switch(config-tablemap)# map from 5 to 46
Switch(config-tablemap)# map from 6 to 56
Switch(config-tablemap)# map from 7 to 57
Switch(config-tablemap)# default 63
```

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

The switch supports a maximum of 256 unique table maps. You can enter up to 64 different **map from–to** entries in a table map. These table maps are supported on the switch:

- DSCP to CoS
- DSCP to precedence
- DSCP to DSCP
- CoS to DSCP
- CoS to precedence
- CoS to CoS
- Precedence to CoS
- Precedence to DSCP
- Precedence to precedence

Table maps modify only one parameter (CoS, IP precedence, or DSCP, whichever is configured) and are only effective when configured with a with a **conform-action** or **exceed-action** command in a police function. Individual policers also support the **violate-action** command.

Table maps are not supported in output policy maps. For more information, see the "Configuring Table Maps" section on page 33-70.

Policing

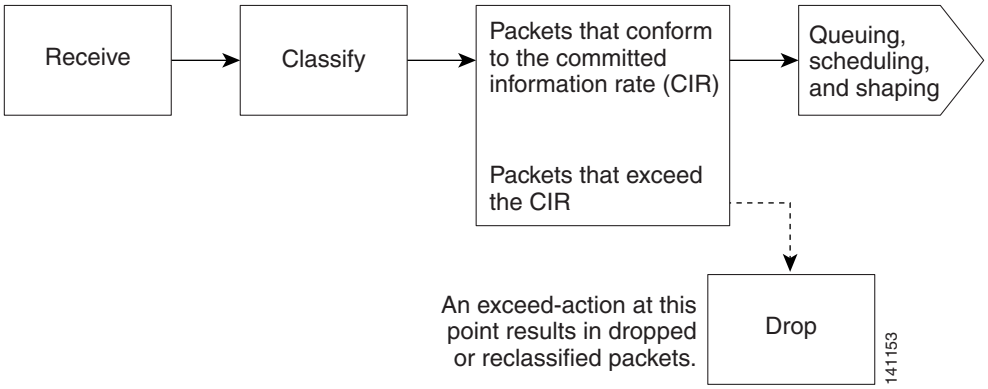
After a packet is classified and assigned a QoS label, you can use policing, as shown in Figure 33-5, to regulate the class of traffic. The policing function limits the amount of bandwidth available to a specific traffic flow or prevents a traffic type from using excessive bandwidth and system resources. A policer identifies a packet as in or out of profile by comparing the rate of the inbound traffic to the configuration profile of the policer and traffic class. 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. The only policing allowed in an output policy map is in priority queues.

All traffic, whether it is bridged or routed, is subjected to a configured policer. As a result, packets might be dropped or might have the DSCP or CoS fields modified when they are policed and marked.

Note Input hierarchical service policies are applied to a traffic stream before any other services act on that traffic. For example, an input hierarchical service policy applied to traffic could change the traffic rate from above a storm-control threshold to below the storm-control threshold, preventing storm control from acting on the traffic stream.

Figure 33-5 Policing of Classified Packets



You can attach a policy map with a policer only in a service policy. The switch supports 1-rate, 2-color and 2-rate, 3-color ingress policing. Only 1-rate, 2-color policing is supported in egress policies.

For 1-rate, 2-color policing, 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 configure the committed information rate (CIR) for updating the first token bucket, and also the peak information rate (PIR) at which the second token bucket is updated. 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 choose to 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.

Setting the burst sizes too low can reduce throughput in situations with bursty traffic. Setting burst sizes too high can allow too high a traffic rate.

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.

Use the **service-policy input** interface configuration command to attach the policy map to a physical port or EFP service instance to make it effective. For more information, see the "Attaching a Service Policy to an Interface or EFP" section on page 33-74. Policing is done only on received traffic, so you can only attach a policer to an input service policy.

See the "Configuring a Policy Map with 1-Rate, 2-Color Policing" section on page 33-43 for configuration examples.

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 an action when the packet conforms to or exceeds the specified

traffic rates. Conform, exceed, and violate actions are to drop the packet, to send the packet without modifications, to set a new CoS, DSCP, or IP precedence value, or to set a QoS group value for classification at the egress.

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

Priority Policing

Priority policing applies only to output policy maps. You can use the **priority** policy map class configuration command in an output policy map to designate a low-latency path or class-based priority queuing for a specific traffic class. With strict priority queuing, the packets in the priority queue are scheduled and sent until the queue is empty. Excessive use of high-priority queuing can create congestion for low priority traffic.

Restriction and Usage Guidelines

- Apply egress policer on priority queues.
- Egress policer is not supported on vlan classes.
- Egress policer is supported at 3rd level or PHB level only on priority queues.
- Conditional marking is not supported in egress policer.
- 1R3C, 2R3C color blind and color aware policer are not supported.
- mefTCM (color-blind/color-aware w/wo coupling) is not supported.
- Policing at Logical or Physical Level and aggregate policing is not supported.
- 2-Level Hierarchical Policing (color-blind/color-aware) is not supported.
- Strict-priority cannot be configured without a policer, if BW is configured on the same level classes.
- Strict priority cannot co-exist with bandwidth kbps/percent in any other class. For strict priority, configure policer first and then configure bandwidth on the same level classes.

To eliminate this congestion, you can use the Priority with Police feature (priority policing) to reduce the bandwidth used by the priority queue, and allocate traffic rates on other queues. Priority with police is the only form of policing supported in output policy maps.

This example shows how to use the **priority** command with **police** command to configure *out-class1* as the priority queue, with traffic going to the queue that is limited to 20,000,000 bps. so that the priority queue never uses more than that. Traffic above that rate is dropped. This allows other traffic queues to receive some port bandwidth, in this case a minimum bandwidth guarantee of 500,000 and 200,000 kbps. The class-default, queue gets the remaining port bandwidth.

```
Switch(config)# policy-map policy1
Switch(config-pmap)# class out-class1
Switch(config-pmap-c)# priority
Switch(config-pmap-c)# police 200000000
Switch(config-pmap-c)# exit
Switch(config-pmap)# class out-class2
Switch(config-pmap-c)# bandwidth 500000
Switch(config-pmap-c)# exit
Switch(config-pmap)# class out-class3
Switch(config-pmap-c)# bandwidth 200000
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# service-policy output policy1
Switch(config-if)# exit
```

Packet Marking

You can use packet marking in input policy maps to set or modify the attributes for traffic belonging to a specific class. For example, you can change the CoS value in a class or set IP DSCP or IP precedence values for a specific type of traffic. These new values are then used to determine how the traffic should be treated. You can also use marking to assign traffic to a QoS group within the switch.

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. This can increase or decrease the priority of a packet in accordance with the policy used in the QoS domain so that other QoS functions can 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.

Restrictions and Usage Guidelines

- Outer Cos, ipv4 dscp, ipv4 precedence, MPLS EXP marking at egress is supported.
- Multiple marking actions for the same class at egress is supported.
- Egress marking of set cos inner is not supported.
- DEI, IPv6 dscp and inner dscp in tunnel is not supported.
- Hierarchical Marking and Enhanced Marking using table-maps is not supported.
- Egress marking cannot be applied to physical classes and vlan classes.
- Qos-group and discard-class marking is not supported at egress, only classification is supported at egress.
- In the case of layer 2 to layer 3 propagation, EXP value is set to 0 by default

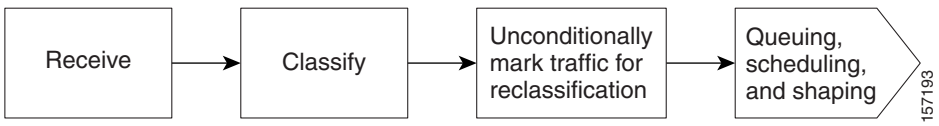
Specify and mark traffic by using the **set** commands in a policy map for all supported QoS markings. CoS, IP DSCP, and IP precedence markings are supported in both ingress and egress policies. The task of setting QoS groups is supported only in ingress policies.

Note The task of setting QoS-groups, discard classes, and imposed EXPs is not supported in egress policies.

A **set** command unconditionally *marks* the packets that match a specific class. You can then attach the policy map to an interface or service instance as an input policy map or output policy map.

You can simultaneously configure the actions to modify the DSCP, precedence, and CoS markings in the packet for the same service along with the QoS group marking actions. You can use the QoS group number defined in the marking action for egress classification. Figure 33-6 shows the steps involved in marking traffic.

Figure 33-6 Marking Classified Traffic



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 gigabitethernet0/1
Switch(config-if)# service-policy input Example
Switch(config-if)# exit
```

If the egress class match criteria is a part of ingress marking set actions, the egress packet will have both ingress and egress marking action applied on it. If ingress and egress set actions act on the same PHB, the egress set action will take precedence over ingress set action.

In this example, the egress packet is set to both DSCP 20 and CoS 3:

```
Switch(config)# policy-map Ingress Example
Switch(config-pmap)# class dscp 10
Switch(config-pmap-c)# set dscp 20
Switch(config-pmap-c)# exit
Switch(config)# policy-map Egress Example
Switch(config-pmap)# class dscp 10
Switch(config-pmap-c)# set cos 5
Switch(config-pmap-c)# exit
```

Congestion Avoidance and Queuing

The Congestion Avoidance feature uses algorithms such as tail drop to control the number of packets entering the queuing and scheduling stage to avoid congestion and network bottlenecks. The switch uses WTD and Weighted Random Early Detection (WRED) to manage the queue sizes and provide a drop precedence for traffic classifications.

These sections contain additional information on congestion avoidance and queuing:

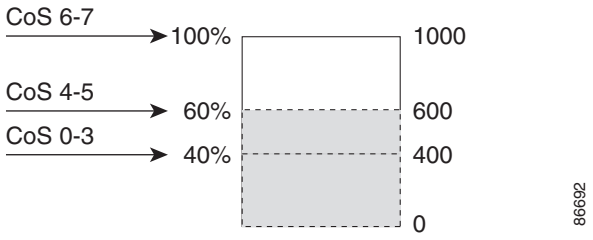
- Weighted Tail Drop, page 33-20
- Weighted Random Early Detection (WRED), page 33-22

Weighted Tail Drop

You set the queue size limits depending on the markings of the packets in the queue. You can assign each packet that travels through the switch to a specific queue and threshold. For example, you can map specific DSCP or CoS values to a specific egress queue and threshold. If the total destination queue size is greater than the threshold of any classified traffic, the next frame of that traffic is dropped.

Figure 33-7 shows an example of WTD operating on a queue with 1000 microseconds worth of data. Three drop percentages are configured: 40 percent (400 microseconds), 60 percent (600 microseconds), and 100 percent (1000 microseconds). These percentages mean that traffic classified to the 40-percent threshold is dropped when the queue depth exceeds 400 microseconds, traffic classified to 60 percent is dropped when the queue depth exceeds 600 microseconds. Traffic up to 400 microseconds can be queued at the 40-percent threshold, up to 600 microseconds at the 60-percent threshold, and up to 1000 microseconds at the 100-percent threshold.

Figure 33-7 WTD and Queue Operation



In this example, CoS values 6 and 7 have a greater importance than the other CoS values, and they are assigned to the 100-percent drop threshold (queue-full state). CoS values 4 and 5 are assigned to the 60-percent threshold, and CoS values 0 to 3 are assigned to the 40-percent threshold.

If the queue is already filled with 600 microseconds worth of data, and a new frame arrives containing CoS values 4 or 5, the frame is subjected to the 60-percent threshold. When this frame is added to the queue, the threshold would be exceeded, so the switch drops it.

You configure WTD by using the **queue-limit** policy-map class command to adjust the queue size (buffer size) associated with a particular class of traffic.

Note Queue-limit is supported only in leaf-level (per-hop behavior) classes.

You specify the maximum threshold in bytes, microseconds, percent or packets. You can specify different queue sizes for different classes of traffic (CoS, DSCP, MPLS EXP, precedence, discard-class, or QoS group) in the same queue. Setting a queue limit establishes a drop threshold for the associated traffic when congestion occurs.

Note You cannot configure queue size by using the **queue-limit** policy map class command without first configuring a scheduling action (**bandwidth**, **shape average**, or **priority**). The only exception to this is when you configure queue-limit for the **class-default** of an output policy map.

The switch supports up to three unique queue-limit configurations (including the default) across all output policy maps. Within an output policy map, four or eight queues (classes) are allowed, including the class default. Each queue can have three defined thresholds. Only three unique threshold value configurations are allowed per class. Multiple policy maps can share the same queue-limits. Each class-map in a policy-map can either share the same threshold or can have its own unique values.

You can use these same queue-limit values in multiple output policy maps on the switch. However, changing one of the queue-limit values in a class creates a new, unique queue-limit configuration. At any one time, you can attach only three unique queue-limit configurations in output policy maps to targets. If you attempt to attach an output policy map with a fourth unique queue-limit configuration, you see this error message:

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

By default, queues have unique thresholds based on the speed of the interface. You can decrease the queue size for latency-sensitive traffic or increase the queue size for bursty traffic.

Queue bandwidth and queue size (queue limit) are configured separately and are not interdependent. You should consider the type of traffic being sent when you configure bandwidth and queue-limit:

- A large buffer (queue limit) can better accommodate bursty traffic without packet loss, but at the cost of increased latency.

- A small buffer reduces latency but is more appropriate for steady traffic flows than for bursty traffic.
- Very small buffers 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 buffer sizes that increase latency are not usually necessary. For high-priority latency-sensitive packets, configure a relatively large bandwidth and relatively small queue size.

WTD thresholds:

- You cannot use the **queue-limit** command to configure more than two threshold values for WTD qualifiers (**cos**, **dscp**, **precedence**, **qos-group, discard-class**, or **mpls experimental**). However, there is no limit to the number of qualifiers that you can map to these thresholds.
- You can configure a third threshold value to set the maximum queue by using the **queue-limit** command with no qualifiers.
- You can configure many more WTD thresholds, provided their threshold values are equal to the maximum threshold of the queue provided by the **queue-limit** command.

See the "Configuring Weighted Tail Drop" section on page 33-64.

Weighted Random Early Detection (WRED)

WRED combines the capabilities of the RED algorithm with the IP Precedence feature to enable preferential traffic handling of high-priority packets. WRED can selectively discard low-priority traffic when the switch begins to get congested, and provide differentiated egress drop thresholds based on the DSCP, IP precedence, CoS values, or MPLS EXP bits.

You can configure WRED to ignore IP precedence when making drop decisions so that non weighted RED behavior is achieved.

WRED attempts to anticipate and avoid congestion rather than control congestion after it occurs.

WRED works with ipv6 dscp/class of service.

Why Use WRED?

WRED facilitates early detection of congestion and enables multiple classes of traffic. It also protects against global synchronization. For these reasons, WRED is useful on any output interface on which you expect congestion to occur.

However, WRED is usually used in the core routers of a network rather than at the edge of a network. Edge routers assign IP precedences to packets as they enter the network. WRED uses these precedences to determine how to treat different types of traffic.

WRED provides separate thresholds and weights for different IP precedences, allowing you to provide different qualities of service with regard to packet dropping for different traffic types. Standard traffic can be dropped more frequently than premium traffic during periods of congestion.

How It Works

By randomly dropping packets prior to periods of high congestion, WRED communicates with the packet source to decrease its transmission rate. If the packet source is using TCP, it will decrease its transmission rate until all the packets reach their destination, which indicates that the congestion is cleared.

WRED generally drops packets selectively, based on IP precedence. Packets with a higher IP precedence are less likely to be dropped than packets with a lower precedence. Thus, the higher the priority of a packet, the higher the probability that the packet will be delivered.

WRED reduces the chances of tail drop by selectively dropping packets when the output policy maps begin to show signs of congestion. By dropping some packets early rather than waiting until the queue is full, WRED avoids dropping large numbers of packets at once and minimizes the chances of global synchronization. Thus, WRED allows the transmission line to be used fully at all times.

In addition, WRED statistically drops more packets from large users than from small users. Therefore, traffic sources that generate the most traffic are more likely to be slowed down than traffic sources that generate less traffic.

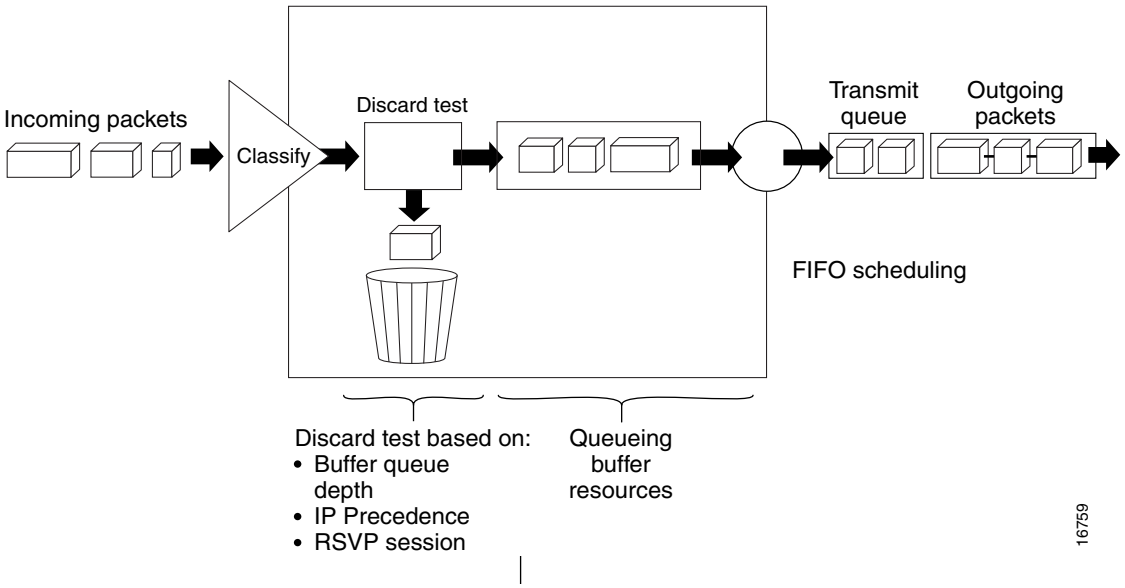
WRED avoids the globalization problems that occur when tail drop is used as the congestion avoidance mechanism. Global synchronization manifests when multiple TCP hosts reduce their transmission rates in response to packet dropping, and increase their transmission rates once again when the congestion is reduced.

WRED is useful only when the bulk of the traffic is TCP/IP traffic. With TCP, dropped packets indicate congestion. Therefore the packet source will reduce its transmission rate. With other protocols, packet sources may not respond or may resend dropped packets at the same rate. Thus, dropping packets does not decrease congestion.

WRED treats non-IP traffic as precedence 0, the lowest precedence. Therefore, non-IP traffic, in general, is more likely to be dropped than IP traffic.

Figure 33-8 illustrates how WRED works.

Figure 33-8 How WRED works.



Average Queue Size

The router automatically determines the parameters to be used in WRED calculations. The average queue size is based on the previous average and the current size of the queue. The formula is:

```
average = (old_average * (1-2 -n)) + (current_queue_size * 2 -n)
```

here *n* is the exponential weight factor, a user-configurable value. The default value of the exponential weight factor is 9. We recommend that you use only the default value for the exponential weight factor. Change this value from the default value only if you have determined that your scenario will benefit from using a different value.

For high values of *n*, the previous average becomes more important. A large factor smooths out the peaks and lows in queue length. The average queue size is unlikely to change very quickly, avoiding drastic swings in size. The WRED process will be slow to start dropping packets, but it may continue dropping packets for a time after the actual queue size has fallen below the minimum threshold. The slow-moving average will accommodate temporary bursts in traffic.

Note If the value of *n* gets too high, WRED will not react to congestion. Packets will be sent or dropped as if WRED were not in effect.

For low values of *n*, the average queue size closely tracks the current queue size. The resulting average may fluctuate with changes in the traffic levels. In this case, the WRED process responds quickly to long queues. After the queue falls below the minimum threshold, the process will stop dropping packets.

If the value of *n* gets too low, WRED will overreact to temporary traffic bursts and drop traffic unnecessarily.

See the "Configuring Weighted Random Early Detection" section on page 33-66.

Congestion Management and Scheduling

MQC provides several related mechanisms in output policy maps to control outgoing traffic flow. The scheduling stage holds packets until the appropriate time to send them to one of the four or eight traffic queues. Queuing assigns a packet to a particular queue based on the packet class and is enhanced by the WTD algorithm for congestion avoidance. 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 that can be consumed by a particular class of traffic and 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 configuration command to specify that a class of traffic should have a maximum permitted average rate. You specify the maximum rate in bits per second.

- 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 bit rate, or as a percentage of total bandwidth, or as remaining bandwidth.

- Priority queuing or class-based low-latency scheduling

  You use the **priority** policy-map class configuration command to specify the that a class of traffic has low latency requirements with respect to other classes. When you configure this command in a class, you cannot configure bandwidth in any other child class associated with the same parent.

These sections contain additional information about scheduling:

- Traffic Shaping, page 33-25
- Class-Based Weighted Fair Queuing, page 33-27
- Priority Queuing, page 33-27

Traffic Shaping

Traffic shaping or class-based peak rate scheduling is used to specify the maximum transmission rate for a traffic class. Unlike traffic policing, where nonconforming packets are dropped or marked right away, packets exceeding the rate specified in the shape command are usually buffered and can be sent later when bandwidth is available. While policing propagates traffic bursts, shaping smooths bursts by sending the packets later. Traffic policing is used in input policy maps, and traffic shaping occurs as traffic leaves an interface or service instance.

Configuring a queue for traffic shaping sets the maximum bandwidth or PIR of the queue. You can set the PIR from 1 Kb/s to 10 Gb/s. You can also configure the PIR as a percentage of the PIR of a parent level.

Note You cannot configure traffic shaping (**shape average**) and or priority queuing (**priority**) for the same class in an output policy map. However, you can configure CIR, PIR, and EIR bandwidth independently for a class and use the **bandwidth**, **bandwidth remaining**, and **shape average** commands at the same time within a class.

Class-Based Shaping

Class-based shaping uses the **shape average** policy-map class configuration command to limit the rate of data transmission in bits per second to be used for the committed information rate for a class of traffic. The switch supports separate queues for four or eight classes of traffic, including the default queue for class **class-default**, unclassified traffic.

See the "Configuring Class-Based Shaping" section on page 33-56.

Port Shaping

Port shaping is applied to all the traffic leaving a target. It uses a policy map with only class default when you specify the maximum bandwidth for a port using the **shape average** command. You can attach a child policy to the class default in a hierarchical policy map format to specify class-based and VLAN-based actions.

Where EFPs are configured to aggregately shape the EFPs in a port, configure a port policy using a class-default shaper configuration.

See the "Configuring Port Shaping" section on page 33-57.

Restriction and Usage Guidelines

- To allow coexistence, apply policy-map on the main interface before applying the policy-map on the sub targets.
- You can configure port level shaping with these policy-maps:
  - Flat policy-map on the service instance.
  - HQoS policy-map on the service instance.
- Port level shaping is supported in the egress direction on the main interface.
- Only Class-default is supported on the port-shaper policy-map.
- EVC QoS is not allowed for user defined classes on the main interface policy-map, or if there is a HQoS policy-map on the main interface.
- Summation of Bandwidth configured on the polices applied on service instances should not exceed the port-shaper value.

- Shape configured on the polices applied on service instance should not exceed the port-shaper value.
- It is recommended to remove the policy-map on the main interface only after removing policy-maps on the service instances.
- It is recommended to apply the policy-map on the main interface before adding the policy-maps on the service instances.
- It is recommended not to change the port-shaper values dynamically when QoS policy-maps are applied.
- Below are the list of unsupported configuration.
  - Port policy(class-default shaper + PHB classes) and policies on EFPs
  - Port policy(class-default shaper + Vlan classes) and policies on EFPs
  - Port policy(class-default shaper + Vlan + PHB classes) and policies on EFPs
  - Port policy(PHB classes) and policies on EFPs
  - Port policy(Vlan classes) and policies on EFPs
  - Port policy(Vlan + PHB classes) and policies on EFPs
  - Port policy(class based policies) to handle LLQ across EFPs + EFP policies
  - Match efp-id on the port policy

Parent-Child Hierarchy

The switch also supports *parent* policy levels and *child* policy levels for traffic shaping. The QoS parent-child structure is used for specific purposes when a child policy is referenced in a parent policy to provide additional control of a specific traffic type.

Note The total of the minimum bandwidth guarantees (CIR) for each queue of the child policy cannot exceed the total port-shape rate.

This is an example of a parent-child configuration:

```
Switch(config)# policy-map parent
Switch(config-pmap)# class class-default
Switch(config-pmap-c)# shape average 50000000
Switch(config-pmap-c)# service-policy child
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitthernet0/1
Switch(config-if)# service-policy output parent
Switch(config-if)# exit
```

You can also configure the PIR in a child policy to be an absolute rate calculated as a percentage of the PIR of the parent level policy. You can configure the child PIR from 0 percent to 100 percent of the parent policy.

```
Switch(config)# policy-map child
Switch(config-pmap)# class class2
Switch(config-pmap-c)# shape average percent 50
Switch(config-pmap-c)# exit
```

Class-Based Weighted Fair Queuing

You can configure class-based weighted fair queuing (CBWFQ) to set the relative precedence of a queue by allocating a portion of the total bandwidth that is available for the port. You use the **bandwidth** policy-map class configuration command to set the minimum guaranteed output bandwidth or CIR for a class of traffic as a rate (kilobits per second), a percentage of total bandwidth, or a percentage of remaining bandwidth.

Note When you configure bandwidth in a policy map, you must configure all rates in the same format, either a configured rate or a percentage. The total of the minimum bandwidth guarantees (CIR) for each queue of the policy cannot exceed the total speed of the parent.

- Configuring bandwidth for a class of traffic as an absolute rate (kilobits per second) or a percentage of total bandwidth represents the minimum bandwidth guarantee (the 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. The CIR range is from 1 Kb/s to 10 Gb/s or 1 to 100 percent. You configure the **bandwidth percent** command mainly in hierarchical policy maps where a child CIR guarantee is tied to the parent CIR guarantee.

  The sum of all CIR commitments for a set of peer classes cannot exceed the PIR (shape) of the parent level. A queue without a configured CIR commitment does not receive any committed bandwidth in the scheduler and can be entirely superseded other classes. If CIR bandwidth is required for any class, including class-default, you must configure it.

  Note You cannot configure bandwidth as an absolute rate or a percentage of total bandwidth when **priority** is configured for another class in the output policy. However, you can configure CIR, EIR, and PIR independently for a class and use the **bandwidth**, **bandwidth remaining**, and **shape average** commands, respectively, at the same time within a class.

- Configuring bandwidth as a percentage of *remaining* bandwidth determines the portion of the excess bandwidth of the target that is allocated to the class. This means that the class is allocated bandwidth only if there is excess bandwidth on the target and if there is no minimum bandwidth guarantee for this traffic class. By default, the total excess bandwidth is divided equally among the classes. You can configure bandwidth as remaining percentage to configure an unequal distribution for prioritizing classes. The total bandwidth that you can allocate between peer classes is 100 percent.

Note You cannot configure bandwidth as percentage of remaining bandwidth when **priority** is configured for another class in the output policy map.

For more information, see the "Configuring Class-Based-Weighted Fair Queuing" section on page 33-53.

Priority Queuing

You can use the **priority** policy-map class configuration command to ensure that a particular class of traffic is given preferential treatment with respect to other classes associated with the same parent entity. With priority queuing, the priority queue is constantly serviced until the queue is empty. With priority queuing, traffic for the associated class is sent before packets in other queues are sent.

When you configure priority in a class, you cannot configure the bandwidth command in any other sibling class associated with the same parent.

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 a single unique class for each policy map.
- 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.

For more information, see the "Configuring Class-Based Priority Queuing" section on page 33-62.

Input and Output Policy Maps

Policy maps are either input policy maps or output policy maps applied to packets as they enter or leave the switch by service policies attached to targets. Input policy maps perform policing and marking on received traffic. Policed packets are 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.

Input policies and output policies have the same basic structure but differ in the characteristics that they regulate. Figure 33-9 shows the relationship of input and output policies.

Figure 33-9 Input and Output Policy Relationship

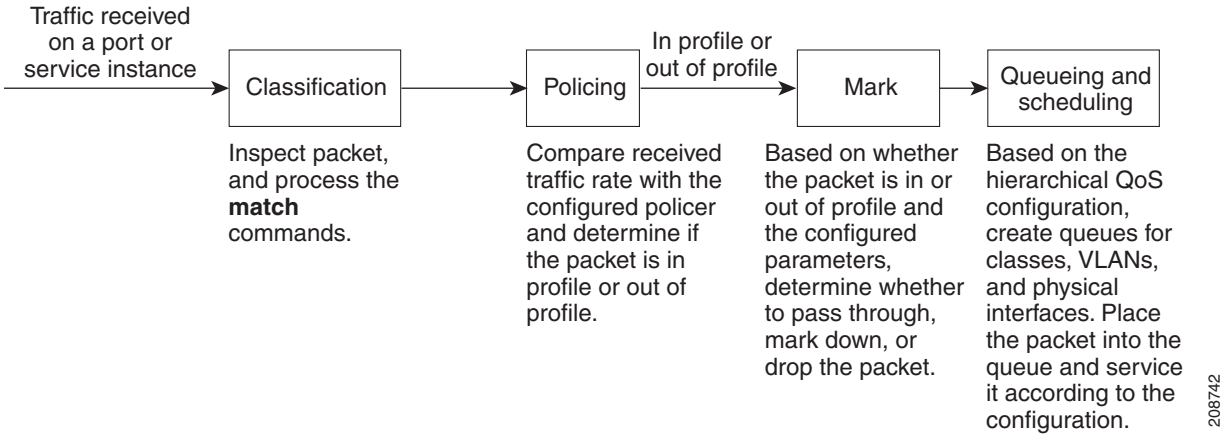


Table 33-1 Options for Input and Output Policies

Action	Feature	Applicable Level
Input Policy Maps		
Classification	Outer VLAN, Inner VLAN, or both	VLAN level.
	Outer CoS, Inner CoS, or both, MAC ACLs, IP ACLs, IPv4 DSCP or Precedence, MPLS EXP Match any (only)	Class level.
	Match any	Physical level or VLAN level.
Marking	Outer CoS, IPv4 DSCP or Precedence, MPLS EXP, QoS group or discard class Multiple marking actions for the same class	Any one level (class, VLAN, or physical)
Policing	1 rate, 2 color 2 rate, 3 color Conditional marking: Outer CoS, IPv4 DSCP or Precedence, MPLS EXP, QoS group or discard class Multiple conditional marking actions.	Any one level (class, VLAN, or physical)
Output Policy Maps		
Classification	Outer VLAN, Inner VLAN, or both	VLAN level
	Outer CoS, Inner CoS, or both, MAC ACLs, IP ACLs, IPv4 DSCP or Precedence, MPLS EXP Match any (only)	Class level
Queuing	Tail drop (queue-limit) or weighted tail drop based on outer CoS, IPv4 DSCP or precedence, MPLS EXP, QoS group or discard class	Class level
Scheduling	Class-based weighted fair queuing (bandwidth)	VLAN or class level.
	Class-based shaping (shape average)	All levels
	Class-based excess bandwidth	VLAN or class level.
	Strict priority	VLAN or class level.

Input Policy Maps

Input policy map classification criteria include matching CoS, DSCP, IP precedence, or MPLS EXP values or matching an ACL or VLAN ID (for per-port, per-VLAN QoS). Input policy maps support two actions: marking and policing. You can specify these actions in any one level of the hierarchical policy

map, but actions cannot be nested. That is, if a child policy has a policer, then the parent policy map cannot have a policer. Likewise, if there is marking in a child policy, there cannot be marking in a parent policy.

Only input policies provide matching on access groups, and only output policies provide matching on QoS groups and discard classes. 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**.

Output Policy Maps

Output policy map classification criteria include matching CoS, DSCP, an IP precedence, MPLS EXP, or QoS groups. or a discard-class value. Output policy maps can have any of these actions:

- Queuing (**queue-limit**)
- Scheduling (**bandwidth**, **priority**, and **shape average**)

Policies attached to an EFP support only 2-level scheduling. Although you can attach a 3-level hierarchical policy to an EFP, the policy should conform to these rules:

- Only two or the three levels can have a scheduling action (bandwidth, priority, or shape).
- One of the two levels must be the class (bottom-most) level.

Output policy maps do not support matching of access groups. You can use QoS groups as an alternative by matching the appropriate access group in the input policy map and setting a QoS group. In the output policy map, you can then match the QoS group. See the "Classification Based on QoS Groups" section on page 33-10 for more information.

- A class-level policy (a child policy with class maps matching CoS, DSCP, IP precedence, MPLS EXP, QoS group, or discard-class) can have a maximum of eight classes including class default.
- A VLAN-level policy (a parent policy that has classes matching VLANs) can have a maximum of 4000 classes, including class default.
- A physical-level policy can have only class-default in the entire policy map.

You can attach an output policy map to any or all targets on the switch. The switch supports configuration and attachment of a unique output policy map for each port or service instance.

However, these output policy maps can contain only three unique configurations of queue limits. These three unique queue-limit configurations can be included in as many output policy maps as there are switch ports. There are no limitations on the configurations of bandwidth, priority, or shaping.

QoS Treatment for Performance-Monitoring Protocols

QoS is not configurable for Cisco IP service level agreements (IP SLA) probes or for traffic to the CPU. QoS treatment is set by default.

Cisco IP-SLAs Probes

For information about Cisco IP service level agreements (IP SLAs), see the "Understanding Cisco IOS IP SLAs" section on page 42-1.

The QoS treatment for IP-SLA probes exactly reflects effects that occur to the normal data traffic crossing the device. The generating device does not change the probe markings. It queues these probes based on the configured queueing policies for normal traffic.

- Marking

  By default, the CoS marking of CFM traffic (including IP SLAs using CFM probes) is not changed. QoS configuration cannot change this behavior.

  By default, IP traffic marking and the CoS marking of all other Layer 2 non-IP traffic is not changed. The QoS marking feature can change this behavior.

- Queuing

  IP SLAs traffic is queued according to its ToS or DSCP value and the output policy map configured on the egress target, similarly to normal traffic. QoS cannot change this behavior.

  By default, all other Layer 2 non-IP traffic and all IP traffic is statically mapped to a queue on the egress target.

CPU Traffic

By default, the switch assigns a separate classifier, *CPU-traffic classifier*, for all traffic destined to the CPU. For traffic from the CPU, the switch uses a default classifier for *high-priority* control protocol traffic. These protocols are considered high priority:

- Protocols with the PAK_PRIORITY flag set in Cisco IOS software. These include EIGHT, HSPR, GRD, LDP, OSPF, RIP, WCCP, BFD, CFM, SAA, CDP, ISIS, DTP, IGRP, Ethernet OAM, LACP, LLDP, PAgP, and STP.
- IP protocols with IP precedence set to 6 or 7.

All other traffic from the CPU is classified with a *normal* classifier.

Configuring QoS

Default QoS Configuration

There are no policy maps, class maps, table maps, or policers configured. At the egress target, all traffic goes through a single default queue that is given the full operational bandwidth.

The packets are not modified. The CoS, DSCP, IP precedence, and MPLS EXP values in the packet are not changed. Traffic is switched in pass-through mode without any rewrites and classified as best effort without any policing.

Configuration Guidelines and Limitations

The ME 3800X and ME 3600X switches support the same QoS functionality. All switches support a total of 4,000 QoS instances, where a QoS instance is any target on which a QoS per-hop behavior policy is attached. A target can be a port, a VLAN class, or an EFP service instance.

- You can configure a maximum of 1024 policy maps.
- You can apply one input policy map and one output policy map to an interface or service instance.
- The maximum number of classification criteria per class map is 64. The maximum number of classes per policy map is 4000.
- Policy configurations are validated as they are configured. When invalid configurations are detected, they are rejected. In some cases the configuration cannot be validated until it is associated with an interface.
- If you modify a feature characteristic on a port or EFP that has a policy map attached and the new configuration makes the policy map invalid, the attached policy is automatically detached.
- You can attach service policies to switchports, routed ports, or EFPs. However, you cannot attach a service policy to a physical port that is configured with service instances (EFPs) and you cannot attach service policies to switch virtual interfaces (SVIs).
- You cannot attach a service policy to an EtherChannel. You can only attach service policies to individual ports in the port channel. You cannot attach a service policy to an EFP that belongs to a port channel interface.
- When a configured policer rate, policer burst-size, or queue-rate cannot be achieved in hardware within 1 percent, the configuration is rejected.
- Egress classification for a class of traffic that has been acted on by an input policy-map can take place only on the per-hop behavior criteria and value established by the input policy-map.
  - If a class in an input policy-map is classifying based on per-hop behavior criteria and is configured for policing but not marking, then the per-hop behavior established by the input policy-map for that class of traffic is the classification criteria and value in the associated class-map.
  - If a class in an input policy-map is not classifying on per-hop behavior criteria, but is classifying on flow criteria by using a MAC ACL or IP ACL, and it is configured for policing, a default *best-effort* per-hop behavior is established for that class of traffic. Traffic in this class is not eligible for egress classification based on per-hop behavior criteria.
  - If a class in an input policy-map is configured for marking, the per-hop behavior established by the input policy-map for that class of traffic is the marked per-hop behavior criteria and value.
  - If a class in an input policy-map is not configured for any action (class-default), a default *best-effort* per-hop behavior is established for that class of traffic. Traffic in this class is not eligible for egress classification based on PHB-criteria.

  For example: if the per-hop behavior established for a class of traffic by an input policy-map is *DSCP ef*, you can classify that class of traffic on the egress based only on *DSCP ef* and not on any other mutually exclusive per-hop behavior such as outer CoS or inner CoS.
- Egress classification on a target with no input policy map attached can use any classification criteria specified in the egress policy. When an input policy map is attached to a target, even if traffic does not match any classes in the input policy, it cannot be egress-classified.
- When configuring both MPLS VPN and QoS, you can apply most QoS functions to MPLS VPN traffic. However, for a hierarchical QoS function, you cannot apply a service policy that would match traffic on a per-VRF basis because VRFs are dynamically assigned to an MPLS label. For

MPLS VPN traffic, you can apply a service-policy on egress traffic that matches traffic based on DSCP or MPLS. For information about configuring QoS with MPLS and EoMPLS, see the "Configuring MPLS and EoMPLS QoS" section on page 33-72.

- There is a limitation per bridge-domain on the number of unique flows classified on inner or outer VLANs that you can configure for egress classification. An error message appears when this limitation is exceeded.
- Hierarchical marking and policing are not supported. You can configure marking and policing for any number of classes on any one of the three levels of the policy-map hierarchy. If you configure marking on one level, you can configure policing without marking (transmit, drop) on another level.
- An EFP can support only 2-level egress scheduling policies. If you attach a 3-level hierarchical policy to an EFP, only 2 of the 3 levels can have scheduling actions (bandwidth, shape, or priority)
- There is a limitation on the maximum number of unique per-hop behavior-criteria combinations that can be configured on the system. The limit is internally calculated and is a function of input classification, marking, and egress classification based on per-hop behavior. An error message appears when this limit is reached.

Configuring Input Policy Maps

Configuring Input Class Maps

You use the **class-map** 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. You define match criteria with one or more **match** statements entered in the class-map configuration mode. Match statements can include criteria such as an ACL, inner or outer CoS value, DSCP value, IP precedence values, MPLS experimental labels, or inner or outer VLAN IDs.

In an input policy, the match criteria acts on the packet on the wire before any VLAN-mapping rewrite operations on ingress.

Beginning in privileged EXEC mode, follow these steps to create a class map and to define the match criterion to classify traffic:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**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. • For *class-map-name*, specify the name of the class map. • (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. **Note** The **match-all** keyword is supported only for outer VLAN and inner VLAN or outer CoS and inner CoS matches for QinQ packets. The **match-all** keyword is rejected for all other mutually exclusive match criteria. • (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.

	Command	Purpose
Step 3	**match** {**access-group** *acl-index-or-name* | **cos** *cos-list* | **cos inner** *cos-list* | **ip dscp** *dscp-list* | **ip precedence** *ip-precedence-list* | **mpls experimental topmost** *value* | **vlan** *vlan-list* | **vlan inner** *vlan-list*}	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. • Enter **cos** *cos-list* to match a packet based on the outer VLAN tag or the service-provider CoS value. You can specify up to four Layer 2 CoS values to match against the packet. Separate each value with a space. The range is 0 to 7. • Enter **cos inner** *cos-list* to match a packet based on the inner CoS value. For packets only one tag, this command has no effect. You can specify up to four Layer 2 CoS values to match against the packet. Separate each value with a space. 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 (af numbers, cs numbers, **default**, or **ef**). • For **ip precedence** *ip-precedence-list*, enter a list of up to four IPv4 precedence values to match against incoming packets. Separate each value with a space. You can enter multiple *ip-precedence-list* lines to match more than four precedence values. The range is 0 to 7. • For **mpls experimental topmost** *value*, enter a list of up to eight MPLS experimental labels. You can enter multiple lines to match more than eight MPLS experimental values. This keyword matches only valid MPLS packets. Separate each value with a space. The range is 0 to 7. • Enter **vlan** *vlan-id* to match a packet based on the outermost, service-provider VLAN ID (S-VLAN). For untagged packets, this matches the default VLAN associated with the packets from the port or EFP. You can enter a single VLAN ID or a range of VLANs separated by a hyphen. The range is from 1 to 4095. • Enter **vlan inner** *vlan-id* to match a packet based on the C-VLAN, the inner customer VLAN ID of an 802.1Q tunnel. For packets with only one tag, the command has no effect. You can specify a single VLAN identified by a VLAN number or a range of VLANs separated by a hyphen. The range is 1 to 4094.
Step 4	**end**	Return to privileged EXEC mode.
Step 5	**show class-map**	Verify your entries.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

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 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
```

This example shows how to create a class map called *class3*, which matches incoming traffic with IP-precedence values of 5, 6, and 7:

```
Switch(config)# class-map match-any class3
Switch(config-cmap)# match ip precedence 5 6 7
Switch(config-cmap)# exit
```

This example shows how to create a class-map called *parent-class*, which matches incoming traffic with VLAN IDs in the range from 30 to 40.

```
Switch(config)# class-map match-any parent-class
Switch(config-cmap)# match vlan 30-40
Switch(config-cmap)# exit
```

Using ACLs to Classify Traffic

You can classify IP traffic by using IP standard or IP extended ACLs. You can classify IP and non-IP traffic by using Layer 2 MAC ACLs. For more information about configuring ACLs, see the chapter on Configuring Network Security with ACLs, page 31-1 in the software configuration guide.

Limitations and Usage Guidelines

The following limitations and usage guidelines apply when classifying traffic using an ACL:

- QoS ACLs are supported only for IPv4 traffic
- QoS ACLs are supported only for ingress traffic
- You can use QoS ACLs to classify traffic based on the following criteria:
  - Source and destination host
  - Source and destination subnet
  - TCP source and destination
  - UDP source and destination
- Named and numbered ACLs are supported.
- You can apply QoS ACLs only to the third level class (bottom-most).
- The following rage of numbered access lists are supported:
  - 1-99—IP standard access list
  - 100-199—IP extended access list
  - 1300-1999—IP standard access list (expanded range)

  - 2000-2699—IP extended access list (expanded range)

- You must create an ACL before referencing it within a QoS policy.
- Deny statements within an ACL are ignored for the purposes of classification.
- Classifying traffic based on TCP flags using an ACL is not supported.
- Classifying traffic using multiple mutually exclusive ACLs within a **match-all** class-map is not supported.
- Classifying traffic on a logical/physical level using an ACL is not supported.
- Applying QoS ACLs to MAC addresses is not supported.
- The **neq** keyword is not supported with the access-list permit and ip access-list extended commands.
- This release does not support matching on multiple port numbers in a single ACE, as in the following command: `permit tcp any eq 23 45 80 any`
- You can only configure 8 port matching operations on a given interface. A given command can consume multiple matching operations if you specify a source and destination port, as shown in the following examples:
  - **permit tcp any lt 1000 any**—Uses one port matching operation
  - **permit tcp any lt 1000 any gt 2000**—Uses two port matching operations
  - **permit tcp any range 1000 2000 any 400 500**—Uses two port matching operations

QoS policies match only the **permit** keyword. Not all ACL criteria are supported for QoS classification. Note the available keywords in the procedures.

To enable layer 4 port matching on the switch use the **platform qos enable layer4-port-match** command.

You can attach a policy that includes unsupported QoS IP ACL options to the target, but QoS ignores the unsupported options. If you modify an IP ACL in a policy map that is already attached to a target and the modification causes the attached policy to become invalid, the policy is detached from the target.

- "Creating IP Standard ACLs" section on page 33-38
- "Creating IP Extended ACLs" section on page 33-38
- "Creating Layer 2 MAC ACLs" section on page 33-40

Creating IP Standard ACLs

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

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**access-list** *access-list-number* **permit** *source* [*source-wildcard*]	Create an IP standard ACL, repeating the command as many times as necessary. • 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.
or	**ip access-list standard** *name*	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*]
Step 3	**end**	Return to privileged EXEC mode.
Step 4	**show access-lists**	Verify your entries.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

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 wildcard 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
```

Creating IP Extended ACLs

Although you can configure many options in ACLs, only some are supported for QoS ACLs.

- For **permit** *protocol*, the supported keywords are: **gre**, **icmp**, **igmp**, **ipinip**, **tcp**, and **udp**.
- For source and destination address, the supported entries are i*p-address*, **any**, or **host**.
- For match criteria, the supported keywords are **dscp** or **tos**. You can also specify a **time-range**.

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

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**access-list** *access-list-number* **permit** *protocol* {*source source-wildcard destination destination-wildcard*} [**tos** *tos*] [**dscp** *dscp*] [**time-range** *name*] **Note** If you enter a **dscp** value, you cannot enter **tos**.	Create an IP extended ACL. Repeat the step as many times as necessary. • 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 *protocol*, enter the name or number of an IP protocol. Although other protocols are visible in the command-line help, only these are supported: IGMP, TCP, UPD, ICMP, IPINIP, and GRE. If you enter other protocol types, the command is rejected. • 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. Although other optional keywords are visible and can be configured, only these are supported in QoS ACLs: • **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. • **time-range**—Specify a configured time range for applying the ACLs. You configure the time range using the **time-range** *time-range-name* global configuration command.
or	**ip access-list extended** *name*	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 source-wildcard destination destination-wildcard*} [**tos** *tos*] [**dscp** *dscp*] [**time-range** *name*] as defined in Step 2.
Step 3	**end**	Return to privileged EXEC mode.
Step 4	**show access-lists**	Verify your entries.
Step 5	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

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 ToS value of 5:

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

Creating Layer 2 MAC ACLs

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

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**mac access-list extended** ***name***	Create a Layer 2 MAC ACL by specifying the name of the list and enter extended MAC ACL configuration mode.
Step 3	**permit** {**any** | **host** *dst-MAC-addr* | *dst-MAC-addr mask*} [*type mask*] **Note** Although visible in the command-line help, the **host** *src-MAC-addr mask* keywords are not supported.	Always use the **permit** keyword for ACLs used as match criteria in QoS policies. • 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. • For *mask,* enter the wildcard bits by placing ones in the bit positions that you want to ignore. • (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. Although other Ethertypes are visible in the command-line help, only IPv4 and MPLS are supported. If you enter another Ethertype, the command is rejected.
Step 4	**end**	Return to privileged EXEC mode.
Step 5	**show access-lists** [*access-list-number* | *access-list-name*]	Verify your entries.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

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 a **permit** statement that allows traffic from the host with MAC address 0001.0000.0001 to the host with MAC address 0002.0000.0001.

```
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)# exit
```

Configuring Class-Based Marking

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

Follow these guidelines when configuring class-based marking:

- Hierarchical marking is not supported.
- You can configure marking for any number of classes on any of the three levels of policy map hierarchy. If you configure marking on one level, you can configure policing without marking (transmit, drop) on another level.

In the privileged EXEC mode, perform these steps to create an input policy map that marks traffic,

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**policy-map** *policy-map-name*	Create a policy map by entering the policy map name, and enter policy-map configuration mode.
Step 3	**class** {*class-map-name* | **class-default**}	Enter a class-map name, or **class-default** to match all unclassified packets, and enter policy-map class configuration mode. If you enter a class-map name, you must have already created the class map by using the **class-map** global configuration command.

	Command	Purpose
Step 4	**set** {**cos** *cos_value* | **discard-class** *value* | [**ip**] **dscp** *dscp_value* | [**ip**] **precedence** *precedence_value* | **mpls experimental** {**imposition** | **topmost**} *experimental _value* | **qos-group** *value*}	Mark traffic by setting a new value in the packet, specifying a table map, or specifying a QoS group. • For **cos** *cos_value,* enter a new CoS value to be assigned to the classified traffic. The range is 0 to 7. • For **discard-class** *value, enter* the exact value to be marked for traffic to be discarded. The range is 0 to 7. • For [**ip**] **dscp** *new-dscp*, enter a new DSCP value to be assigned to the classified traffic. The range is 0 to 63 or valid af numbers, cs numbers, **default**, or **ef**. • For [**ip**] **precedence** *new-precedence,* enter a new IP-precedence value to be assigned to the classified traffic, 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**). **Note** A class can have either DSCP or precedence marking. If one of these is already configured in a class and you configure the other keyword, the newer command overwrites the previous command. If the value configured for a **set ip precedence** class in an earlier class in a policy overlaps the value configured for a **set ip dscp** class in a later class, then the earlier configuration is always matched. • For **mpls experimental imposition** *exp-number*, enter the new MPLS experimental value to be set at tag imposition. This keyword specifies that the MPLS experimental value in a packet header is set with the new value after the packet is switched. This keyword applies only to MPLS packets that are MPLS routed. The range is 0 to 7. • **mpls experimental topmost** *exp-number*, enter the new MPLS experimental value for the outermost or topmost label. This keyword marks all valid MPLS packets. The range is 0 to 7. • For **qos-group** *value,* identify a QoS group to be used at egress to identify specific packets. The range is from 0 to 99. If the policy is already attached to an interface, you must exit policy-map class configuration mode before the modification is applied.
Step 5	**end**	Return to privileged EXEC mode.
Step 6	**show policy-map** [*policy-map-name* [**class** *class-map-name*]]	Verify your entries.
Step 7	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

After you create the input policy map, attach it to a target. See the "Attaching a Service Policy to an Interface or EFP" section on page 33-74. Use the **no** form of the appropriate command to delete a policy map or remove an assigned CoS, DSCP, MPLS, precedence, or QoS-group 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 gigabitethernet 0/1
Switch(config-if)# service-policy input Example
Switch(config-if)# exit
```

Configuring Policing

Policing is used to enforce a traffic-metering profile. A policer meters a particular traffic flow and determines if a packet in the given flow conforms to the specified rate. You 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.

The switch supports 1-rate policing with a 2-color marker, or 2-rate policing with a 3-color marker. Mapped packets can be sent without modification, dropped, or marked to the options specified by the **set** command. Note that traffic rates are configured in bits per second and burst size is entered in bytes.

Follow these guidelines when configuring policing.

- Hierarchical policing is not supported.
- You can configure policing for any number of classes on any one of the three levels of the policy-map hierarchy. If you configure marking on one level, you can configure policing without marking (transmit, drop) on another level.

Configuring a Policy Map with 1-Rate, 2-Color Policing

Beginning in privileged EXEC mode, follow these steps to create a 1-rate, 2 color input policy map with individual policing:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**class** {*class-map-name* | **class-default**}	Enter a class-map name or **class-default** to match all unclassified packets, and enter policy-map class configuration mode. If you enter a class-map name, you must have already created the class map by using the class-map global configuration command.
Step 3	**policy-map** *policy-map-name*	Create a policy map by entering the policy map name, and enter policy-map configuration mode. By default, no policy maps are defined.

Command	Purpose
Step 4 **police** {*rate-bps* | **cir** {*cir-bps* [*burst-bytes*] [**bc-***burst-bytes*]| **percent** *percent* [*burst-ms*] [**bc-***burst-ms*]} }	Configure a traffic policer based on the traffic rate or committed information rate (CIR). By default, no policer is defined. • For *rate-bps,* specify average traffic rate in bits per second (b/s). The range is 64000 to 10000000000. • For **cir** *cir-bps,* specify a committed information rate (CIR) in bits per second (b/s). The range is 64000 to 10000000000. • For *burst-bytes* (optional), specify the normal burst size in bytes. The range is 8000 to 16000000. • For **bc-***burst-bytes* (optional), specify the conformed burst (bc) or the number of acceptable burst bytes. The range is 8000 to 16000000. • For **cir percent** *percent*, specify the rate as a percentage of the bandwidth assigned to the class. The range is 1 to 100 percent. • For *burst-ms* (optional), enter the conform burst size in milliseconds. The range is 1 to 2000. The default is 250 ms. • For **bc-***burst-ms* (optional), specify the conformed burst (bc) in milliseconds. The range is 1 to 2000. **Note** If you are configuring a single action for conformed and exceeded packets, you can specify them in the same line as the **police** command. If configuring multiple actions, press ENTER after the **police** command, and enter policy-map class police configuration mode (config-pmap-c-police) mode to specify the actions to take.

	Command	Purpose
Step 5	**conform-action** {**drop** | **set-cos-transmit** *cos_value* | **set-discard-class** *discard_value* | **set-dscp-transmit** *dscp_value* | **set-mpls-exp-imposition-transmit** *new-exp* | **set-mpls-exp-topmost-transmit** *new-exp* | **set-prec-transmit** *value* | **set-qos-transmit** *value* | **transmit**}	(Optional) Enter the action to be taken on packets that conform to the CIR. • **drop**—drop the packet. • **set-cos-transmit** *cos_value*—set the CoS value to a new value, and send the packet. The range is 0 to 7. • **set-discard-class-transmit** *discard_value*—set the discard value to a new value, and send the packet. The range is 0 to 7. • **set-dscp-transmit** *dscp_value*—set the IP DSCP value to a new value, and send the packet. The range is 0 to 63. You also can enter a mnemonic name for a commonly used value or use the question mark (?) to see a list of available values. • **set-mpls-exp-imposition-transmit** *new-exp*—enter the new MPLS experimental value to be set at tag imposition, and send the packet. The range is 0 to 7. • **set-mpls-exp-topmost-transmit** *new-exp*—enter the new MPLS experimental value for the outermost or topmost label, and send the packet. The range is 0 to 7. • **set-prec-transmit** *value*—set the IP precedence value to a new value, and send the packet. The range is 0 to 7. • **set-qos-transmit value**—set the QoS group number to a new value, and send the packet. The range is 0 to 99. • **transmit**—send the packet without altering it. This is the default. **Note** If you are configuring a single action for conformed and exceeded packets, you can specify them in the same line. If configuring multiple actions, press ENTER after the **conform-action** command.

	Command	Purpose
Step 6	**exceed-action** {**drop** | **set-cos-transmit** *cos_value* | **set-discard-class** *discard_value* | **set-dscp-transmit** *dscp_value* | **set-mpls-exp-imposition-transmit** *new-exp* | **set-mpls-exp-topmost-transmit** *new-exp* | **set-prec-transmit** *value* | **set-qos-transmit** *value* | **transmit**}	(Optional) Enter the action to be taken on packets that exceed the CIR. The default exceed action, if no action is configured, is **drop**. • **drop**—drop the packet. • **set-cos-transmit** *cos_value*—set the CoS value to a new value, and send the packet. The range is 0 to 7. • **set-discard-class-transmit** *discard_value*—set the discard value to a new value, and send the packet. The range is 0 to 7. • **set-dscp-transmit** *dscp_value*—set the IP DSCP value to a new value, and send the packet. The range is 0 to 63. You also can enter a mnemonic name for a commonly used value. • **set-mpls-exp-imposition-transmit** *new-exp*—enter the new MPLS experimental value to be set at tag imposition. The range is 0 to 7. • **set-mpls-exp-topmost-transmit** *new-exp*—enter the new MPLS experimental value for the outermost or topmost label, and send the packet. The range is 0 to 7. • **set-prec-transmit** *value*—set the IP precedence value to a new value, and send the packet. The range is 0 to 7. • **set-qos-transmit** *value*—set the QoS group number to a new value, and send the packet. The range is 0 to 99. • **transmit**—send the packet without altering it. **Note** If you explicitly configure **exceed-action drop** as keywords in the command, you must enter policy-map class police configuration mode and enter the **no exceed-action drop** command to remove the previously configured exceed-action before you can enter the new exceed-action.
Step 7	**end**	Return to privileged EXEC mode.
Step 8	**show policy-map** [*policy-map-name* [**class** *class-map-name*]]	Verify your entries.
Step 9	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

After you create the policy map, attach it to an interface or an EFP. See the "Attaching a Service Policy to an Interface or EFP" section on page 33-74.

This example shows how to create a traffic classification with a CoS value of 4, create a policy map, and attach it to an ingress port. The average traffic rate is limited to 10000000 b/s with a burst size of 10000 bytes:

```
Switch(config)# class-map video-class
Switch(config-cmap)# match cos 4
Switch(config-cmap)# exit
Switch(config)# policy-map video-policy
Switch(config-pmap)# class video-class
Switch(config-pmap-c)# police 10000000 10000
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# service-policy input video-policy
Switch(config-if)# exit
```

This example shows how to create policy map with a conform action of **set dscp** and a default exceed action, and attach it to an EFP.

```
Switch(config)# class-map in-class-1
Switch(config-cmap)# match dscp 14
Switch(config-cmap)# exit
Switch(config)# policy-map in-policy
Switch(config-pmap)# class in-class-1
Switch(config-pmap-c)# police 230000 8000 conform-action set-dscp-transmit 33
exceed-action drop
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet0/1
Switch (config-if)# service instance 1 Ethernet
Switch (config-if-srv)# service-policy input in-policy
Switch (config-if-srv)# exit
```

This example shows how to use policy-map class police configuration mode to set multiple conform actions and an exceed action. The policy map sets a committed information rate of 23000 bits per second (b/s) and a conform burst size of 10000 bytes. The policy map includes multiple conform actions (for DSCP and for Layer 2 CoS) and an exceed action.

```
Switch(config)# class-map cos-set-1
Switch(config-cmap)# match cos 3
Switch(config-cmap)# exit
Switch(config)# policy-map map1
Switch(config-pmap)# class cos-set-1
Switch(config-pmap-c)# police cir 23000 bc 10000
Switch(config-pmap-c-police)# conform-action set-dscp-transmit 48
Switch(config-pmap-c-police)# conform-action set-cos-transmit 5
Switch(config-pmap-c-police)# exceed-action drop
Switch(config-pmap-c-police)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# service-policy input map1
Switch(config-if)# exit
```

Configuring a Policy Map with 2-Rate, 3-Color Policing

A 2-rate, 3-color policer uses both the committed information rate (CIR) and the peak information rate (PIR) and includes three profiles (colors) for packets that conform, exceed, or violate the specified rates. You can configure actions to take on each of these profiles.

Beginning in privileged EXEC mode, follow these steps to create an input policy map with individual 2-rate, 3-color policing:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**class** {*class-map-name* | **class-default**}	Enter a class-map name or **class-default** to match all unclassified packets, and enter policy-map class configuration mode. If you enter a class-map name, you must have already created the class map by using the **class-map** global configuration command.
Step 3	**policy-map** *policy-map-name*	Create a policy map by entering the policy map name, and enter policy-map configuration mode. By default, no class maps are defined.

	Command	Purpose
Step 4	**police** {*rate-bps* | **cir** {*cir-bps* [*burst-bytes*] [**bc** *burst-bytes*] | **percent-***percent* [*burst-ms*] [**bc** *burst-ms*]} [**pir** {*pir-bps* [**be** *peak-burst*] | **percent** *percent* [**be** *peak-ms*]}	Define a policer using one or two rates—committed information rate (CIR) and peak information rate (PIR) for the class of traffic. By default, no policer is defined. • For *rate-bps,* specify average traffic rate in bits per second (b/s). The range is 64000 to 10000000000. • For **cir** *cir-bps,* specify a committed information rate (CIR) in bits per second (b/s). The range is 64000 to 10000000000. • For *burst-bytes* (optional), specify the normal burst size in bytes. The range is 8000 to 16000000. • For **bc-***burst-bytes* (optional), specify the conformed burst (bc) or the number of acceptable burst bytes. The range is 8000 to 16000000. • For **cir percent** *percent*, specify the CIR as a percentage of the bandwidth assigned to the class. The range is from 1 to 100 percent. • For *burst-ms* (optional), enter the conform burst size in milliseconds. The range is 1 to 2000. • For **bc-***burst-ms* (optional), specify the conformed burst (bc) in ms. The range is 1 to 2000. • (Optional) For **pir** *pir-bps,* specify the peak information rate (PIR) for the policy. The range is 64000 to 10000000000. If you do not enter a **pir** *pir-bps*, the policer is configured as a 1-rate, 2-color policer. • For **be** *peak-burst* (optional), specify the peak burst size in bytes. The range is 8000 to 16000000 bytes. The default is internally calculated based on the user configuration. You cannot configure this option unless you have entered the **pir** keyword. • For **pir percent** *percent*, specify the PIR as a percentage of the bandwidth assigned to the class. The range is 1 to 100 percent. if you enter cir percent, you must enter pir in percent. • For **be** *burst-ms* (optional), specify the peak burst in ms. The range is 1to 2000. **Note** If you are configuring a single action for conformed and exceeded packets, you can specify them in the same line as the **police** command. If configuring multiple actions, press ENTER after the **police** command, and enter policy-map class police configuration mode (config-pmap-c-police) mode to specify the actions to take.

	Command	Purpose
Step 5	**conform-action** {**drop** | **set-cos-transmit** *cos_value* | **set-discard-class -transmit** *discard_value* | **set-dscp-transmit** *dscp_value* | **set-mpls-exp-imposition-transmit** *new-exp* | **set-mpls-exp-topmost transmit** *new-exp* | **set-prec-transmit** *value* | **set-qos-transmit** *value* | **transmit**}	(Optional) Enter the action to be taken on packets, depending on whether or not they conform to the CIR and PIR. • (Optional) For **conform-action**, specify the action to perform on packets that conform to the CIR and PIR. The default is **transmit**. • (Optional) For **exceed-action**, specify the action to perform on packets that conform to the PIR but not the CIR. The default is **drop**. • (Optional) For **violate-action**, specify the action to perform on packets that exceed the PIR. This keyword is only visible when you have entered **pir** after the **police** command. The default is **drop**.
	| **exceed-action** {**drop** | **set-cos-transmit** *cos_value* | **set-discard-class -transmit** *discard_value* | **set-dscp-transmit** *dscp_value* | **set-mpls-exp-imposition-transmit** *new-exp* | **set-mpls-exp-topmost transmit** *new-exp* | **set-prec-transmit** *value*+ **set-qos-transmit** *value*+**transmit**}	Specify one of these actions to perform on the packets: • **drop**—drop the packet. **Note** If the conform action is set to **drop**, the exceed and violate actions are automatically set to **drop**. If the exceed action is set to **drop**, the violate action is automatically set to **drop**. • **set-cos-transmit** *cos_value*—set the CoS value to a new value, and send the packet. The range is 0 to 7. • **set-discard-class -transmit** *discard_value*—set the discard value to a new value, and send the packet. The range is 0 to 7. • **set-dscp-transmit** *dscp_value*—set the IP DSCP value to a new value, and send the packet. The range is 0 to 63. You also can enter a mnemonic name for a commonly used value. • **set-mpls-exp-imposition-transmit** *new-exp*—enter the new MPLS experimental value to be set at tag imposition, and send the packet. The range is 0 to 7. • **set-mpls-exp-topmost-transmit** *new-exp*—enter the new MPLS experimental value for the outermost or topmost label, and send the packet. The range is 0 to 7. • **set-prec-transmit** *value*—set the IP precedence value to a new value, and send the packet. The range is 0 to 7. • **set-qos-transmit** *value*—set the QoS group number to a new value, and send the packet. The range is 0 to 99. • **transmit**—send the packet without altering it. **Note** You can enter a single conform-action, exceed-action, or violate-action as part of the command string following the police command. You can also press Enter after the **police** command to enter policy-map class police configuration mode, where you can enter multiple actions. In policy-map class police configuration mode, you must enter an action to take.
	| **violate- action** {**drop** | **set-cos-transmit** *cos_value* | **set-discard-class -transmit** *discard_value* | **set-dscp-transmit** *dscp_value* | **set-mpls-exp-imposition-transmit** *new-exp* | **set-mpls-exp-topmost transmit** *new-exp* | **set-prec-transmit** *value* | **set-qos-transmit** *value* | **transmit**}	
Step 6	**end**	Return to privileged EXEC mode.
Step 7	**show policy-map** [*policy-map-name*]	Verify your entries.
Step 8	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

After you create the policy map, attach it to an interface or an EFP. See the "Attaching a Service Policy to an Interface or EFP" section on page 33-74.

Use the **no** form of the appropriate command to delete an existing policy map, class map, or policer.

This example shows how to configure 2-rate, 3-color policing using policy-map configuration mode.

```
Switch(config)# class-map cos-4
Switch(config-cmap)# match cos 4
Switch(config-cmap)# exit
Switch(config)# policy-map in-policy
Switch(config-pmap)# class cos-4
Switch(config-pmap-c)# police cir 5000000 pir 8000000 conform-action transmit
exceed-action set-dscp-transmit 24 violate-action drop
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# service-policy input in-policy
Switch(config-if)# exit
```

This example shows how to create the same configuration using policy-map class police configuration mode.

```
Switch(config)# class-map cos-4
Switch(config-cmap)# match cos 4
Switch(config-cmap)# exit
Switch(config)# policy-map in-policy
Switch(config-pmap)# class cos-4
Switch(config-pmap-c)# police cir 5000000 pir 8000000
Switch(config-pmap-c-police)# conform-action transmit
Switch(config-pmap-c-police)# exceed-action set-dscp-transmit 24
Switch(config-pmap-c-police)# violate-action drop
Switch(config-pmap-c-police)# end
```

Configuring Output Policy Maps

Configuring Output Class Maps

You 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 inner or outer CoS value, DSCP value, IP precedence values, QoS group values, discard class, MPLS experimental labels, or inner or outer VLAN IDs. You define match criterion with one or more **match** statements entered in the class-map configuration mode.

In an output policy, the match criteria acts on the packet on the wire after any VLAN rewrite mapping operations on egress.

Beginning in privileged EXEC mode, follow these steps to create a class map and to define the match criterion to classify traffic:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**class-map** [**match-all** | **match-any**] *class-map-name* **Note** The **match-all** keyword is supported only for outer VLAN and inner VLAN, or outer CoS and inner CoS matches for QinQ packets. The **match-all** keyword is rejected for all other mutually exclusive match criteria.	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) except to match outer and inner 802.1Q VLAN tag of QinQ packets using **match vlan** and **match vlan inner** or **match cos** and **match cos inner.**

	Command	Purpose
Step 3	**match** {**cos-***cos-list* | **cos inner** *cos-list* | **discard-class-***value* | **ip dscp** *dscp-list* | **ip precedence** *ip-precedence-list* | **mpls experimental topmost** *value*| **qos-group** *value* | **vlan-***vlan-list* | **vlan inner** *vlan-list*}	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. • Enter **cos** *cos-list* to match a packet based on the outer VLAN tag or the service-provider CoS value. You can specify up to four Layer 2 CoS values to match against the packet. Separate each value with a space. The range is 0 to 7. • Enter **cos inner** *cos-list* to match a packet based on the inner CoS value. For packets with less than two tags, this command has no effect. You can specify up to four Layer 2 CoS values to match against the packet. Separate each value with a space. The range is 0 to 7. • Enter **discard-class-***value* to match a packet based the drop precedence for a packet during congestion management. The range is 0 to 7. Matching discard class is supported only in output policy maps • 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 (af numbers, cs numbers, **default**, or **ef**). • For **ip precedence** *ip-precedence-list*, enter a list of up to four IPv4 precedence values to match against incoming packets. Separate each value with a space. You can enter multiple *ip-precedence-list* lines to match more than four precedence values. The range is 0 to 7. • For **mpls experimental topmost** *value*, enter a list of up to eight MPLS experimental labels. You can enter multiple lines to match more than eight MPLS experimental values. This keyword matches only valid MPLS packets. Separate each value with a space. The range is 0 to 7. • For **qos-group** *value,* specify the QoS group number. The range is 0 to 99. Matching of QoS groups is supported only in output policy maps. • Enter **vlan** *vlan-id* to match a packet based on the outermost, service-provider VLAN ID (S-VLAN). For untagged packets, this matches the default VLAN associated with the packets from the port or EFP. You can enter a single VLAN ID or a range of VLANs separated by a hyphen. The range is from 1 to 4095. • Enter **vlan inner** *vlan-id* to match a packet based on the C-VLAN, the inner customer VLAN ID of an 802.1Q tunnel. For packets with less than 2 tags, the command has no effect. You can specify a single VLAN identified by a VLAN number or a range of VLANs separated by a hyphen. The range is 1 to 4094.
Step 4	**end**	Return to privileged EXEC mode.
Step 5	**show class-map**	Verify your entries.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

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 a class map called *class2*, which matches 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
```

This example shows how to create a class map called *class3*, which matches traffic with IP-precedence values of 5, 6, and 7:

```
Switch(config)# class-map match-any class3
Switch(config-cmap)# match ip precedence 5 6 7
Switch(config-cmap)# exit
```

This example shows how to create a class-map called *parent-class*, which matches traffic with VLAN IDs in the range from 30 to 40.

```
Switch(config)# class-map match-any parent-class
Switch(config-cmap)# match vlan 30-40
Switch(config-cmap)# exit
```

Configuring Class-Based-Weighted Fair Queuing

You use the **bandwidth** policy-map class configuration command to configure class-based weighted fair queuing (CBWFQ). CBWFQ sets the explicit minimum guaranteed rate (CIR) of a class by reserving the configured bandwidth for that class.

Follow these guidelines:

- You can configure CBWFQ at the class level and at the VLAN level.
- The total of the minimum bandwidth guaranteed for each queue of the policy cannot exceed the total speed of the parent.
- You cannot configure bandwidth as an absolute rate or a percentage of total bandwidth when strict priority is configured for another class in the output policy.
- You can configure bandwidth as percentage of remaining bandwidth when strict priority is configured for another class in the output policy map.
- You cannot configure bandwidth and priority or bandwidth and traffic shaping for the same class in an output policy map.

Beginning in privileged EXEC mode, 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:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**policy-map** *policy-map-name*	Create a policy map by entering the policy map name, and enter policy-map configuration mode.
Step 3	**class** {*class-map-name* | **class-default**}	Enter a *child class-map name* or **class-default** to match all unclassified packets, and enter policy-map class configuration mode.

	Command	Purpose
Step 4	**bandwidth** {*rate* | **percent** *value* | **remaining percent** *value*}	Set output bandwidth limits for the policy-map class. • Enter a *rate* to set bandwidth in kilobits per second. The range is from 1 to 100000000. • Enter **percent** *value* to set bandwidth as an absolute 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 0 to 100 percent. The total guaranteed bandwidth cannot exceed the total available rate.
Step 5	**end**	Return to privileged EXEC mode.
Step 6	**show policy-map** [*policy-map-name* [**class** *class-map-name*]]	Verify your entries.
Step 7	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

After you create the policy map, attach it to an interface or an EFP. See the "Attaching a Service Policy to an Interface or EFP" section on page 33-74. Use the **no** form of the appropriate command to delete an existing policy map, class map, or bandwidth configuration.

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 GigabitEthernet0/1
```

This example shows how to allocate 25 percent of the total available bandwidth to the traffic class defined by the class map:

```
Switch(config)# policy-map gold_policy
Switch(config-pmap)# class out_class-1
Switch(config-pmap-c)# bandwidth percent 25
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# service-policy output gold_policy
Switch(config-if)# exit
```

Note When you configure CIR bandwidth for a class as an absolute rate or 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 not eligible for any excess bandwidth and, as a result, receives no bandwidth.

This example shows how to set the precedence of output queues by setting bandwidth in kilobits per second. The classes *outclass1*, *outclass2*, and *outclass3* and **class-default** get a minimum of 40000, 20000, 10000, and 10000 kb/s. Any excess bandwidth is divided among the classes in the same proportion as the CIR rate.

```
Switch(config)# policy-map out-policy
Switch(config-pmap)# class outclass1
Switch(config-pmap-c)# bandwidth 40000
Switch(config-pmap-c)# exit
Switch(config-pmap)# class outclass2
Switch(config-pmap-c)# bandwidth 20000
Switch(config-pmap-c)# exit
Switch(config-pmap)# class outclass3
Switch(config-pmap-c)# bandwidth 10000
Switch(config-pmap-c)# exit
Switch(config-pmap)# class class-default
Switch(config-pmap-c)# bandwidth 10000
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet 0/1
Switch(config-if)# service-policy output out-policy
Switch(config-if)# exit
```

This example shows how to allocate the excess bandwidth among queues by configuring bandwidth for a traffic class as a percentage of remaining bandwidth. The class *outclass1* is given priority queue treatment. The other classes are configured to get percentages of the excess bandwidth if any remains after servicing the priority queue: *outclass2* is configured to get 50 percent, *outclass3* to get 20 percent, and the class **class-default** to get the remaining 30 percent.

```
Switch(config)# policy-map out-policy
Switch(config-pmap)# class outclass1
Switch(config-pmap-c)# priority
Switch(config-pmap-c)# exit
Switch(config-pmap)# class outclass2
Switch(config-pmap-c)# bandwidth remaining percent 50
Switch(config-pmap-c)# exit
Switch(config-pmap)# class outclass3
Switch(config-pmap-c)# bandwidth remaining percent 20
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet 0/1
Switch(config-if)# service-policy output out-policy
Switch(config-if)# exit
```

This example shows how to match VLAN and CoS in the same policy. When you attach the service policy *vlan* to an interface, packets with the outer VLAN of 2 and an outer CoS of 2 are included in class map *phb*.

```
Switch(config)# class-map vlan
Switch(config-cmap)# match vlan 2
Switch(config-cmap)# exit
Switch(config)# class-map phb
Switch(config-cmap)# match cos 2
Switch(config-cmap)# exit
Switch(config)# policy-map phb
Switch(config-pmap)# class phb
Switch(config-pmap-c)# bandwidth 1000
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# policy-map vlan
Switch(config-pmap)# class vlan
Switch(config-pmap-c)# bandwidth 1000
Switch(config-pmap-c)# service-policy phb
```

```
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet 0/1
Switch(config-if)# service-policy vlan
Switch(config-if)# exit
```

Configuring Class-Based Shaping

You use the **shape average** policy-map class configuration command to configure traffic shaping. Class-based shaping sets the explicit maximum peak information rate (PIR) for the class by limiting it to the configured bandwidth. You can configure class-based shaping at the class level and at the VLAN level.

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

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**policy-map** *policy-map-name*	Create a policy map by entering the policy map name, and enter policy-map configuration mode.
Step 3	**class** {*class-map-name* | **class-default**}	Enter a *child class-map name* or **class-default** to match all unclassified packets, and enter policy-map class configuration mode.
Step 4	**shape average** {*target bps* | **percent** *value*}	Specify the average class-based shaping rate. • For *target bps*, specify the average bit rate in bits per second. The range is from 1000 to 10000000000 (10 Gigabits). • Enter **percent** *value* to set the percentage of interface bandwidth for peak information rate. The range is 0 to 100 percent. The percentage is calculated based on the peak information rate (PIR) of the parent class. If there is no configured PIR at any level, this is the percentage of the interface speed. Setting the percent to 0 disables shaping.
Step 5	**end**	Return to privileged EXEC mode.
Step 6	**show policy-map** [*policy-map-name* [**class** *class-map-name*]]	Verify your entries.
Step 7	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

After you create the policy map, attach it to an interface or an EFP. See the "Attaching a Service Policy to an Interface or EFP" section on page 33-74. Use the **no** form of the appropriate command to delete an existing policy map or class map or to delete a class-based shaping configuration.

This example shows how to configure traffic shaping for outgoing traffic on a Gigabit Ethernet port so that *outclass1*, *outclass2*, and *outclass3* get a maximum of 50, 20, and 10 Mb/s of the available port bandwidth.

```
Switch(config)# policy-map out-policy
Switch(config-pmap)# class classout1
Switch(config-pmap-c)# shape average 50000000
Switch(config-pmap-c)# exit
Switch(config-pmap)# class classout2
Switch(config-pmap-c)# shape average 20000000
Switch(config-pmap-c)# exit
```

```
Switch(config-pmap)# class classout3
Switch(config-pmap-c)# shape average 10000000
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# service-policy output out-policy
Switch(config-if)# exit
```

Configuring Port Shaping

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. A child policy can be attached to the class-default in a hierarchical policy map format to specify class-based and VLAN-based actions.

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

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**policy-map** *policy-map-name*	Create a hierarchical policy map by entering the hierarchical policy map name, and enter policy-map configuration mode for the parent policy.
Step 3	**class class-default**	Enter a policy-map class configuration mode for the default class.
Step 4	**shape average** {*target bps* | **percent** *value*}	Specify the average class-based shaping rate. • For *target bps*, specify the average bit rate in bits per second. The range is from 1000 to 10000000000 (10 Gigabits). • Enter **percent** *value* to set the percentage of interface bandwidth for peak information rate. The range is 0 to 100 percent. The percentage is based on the port operational link speed. Setting the percent to 0 disables shaping.
Step 5	**service-policy** *policy-map-name*	Specify the child policy-map to be used in the hierarchical policy map if required.
Step 6	**end**	Return to privileged EXEC mode.
Step 7	**show policy-map** [*policy-map-name* [**class** *class-map-name*]]	Verify your entries.
Step 8	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

After you create the policy map, attach it to an interface or an EFP. See the "Attaching a Service Policy to an Interface or EFP" section on page 33-74.

Use the **no** form of the appropriate command to delete an existing hierarchical policy map, to delete a port shaping configuration, or to remove the policy map from the hierarchical policy map.

This example shows how to configure port shaping by configuring a hierarchical policy map that shapes a port to 90 Mb/s, allocated according to the *out-policy* policy map configured in the previous example.

```
Switch(config)# policy-map out-policy-parent
Switch(config-pmap)# class class-default
Switch(config-pmap-c)# shape average 90000000
Switch(config-pmap-c)# service-policy out-policy
Switch(config-pmap-c)# exit
```

```
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# service-policy output out-policy-parent
Switch(config-if)# exit
```

Configuring Marking

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

Follow these guidelines when configuring class-based marking:

- Hierarchical marking is not supported.
- You can only configure egress marking for classes on the third level of the policy map hierarchy.

In the privileged EXEC mode, perform these steps to create an output policy map that marks traffic.

	Command	Purpose
Step 1	**configure terminal**	Enter the global configuration mode.
Step 2	**policy-map** *policy-map-name*	Create a policy map by entering the policy map name and enter the policy-map configuration mode.
Step 3	**class** {*class-map-name* | **class-default**}	Enter a class map name or the **class-default** command to match all unclassified packets and enter policy-map class configuration mode. To enter a class map name, you must have already created the class map by using the **class-map** global configuration command.
Step 4	**set** {**cos** *cos_value* | **discard-class** *value* | [**ip**] **dscp** *dscp_value* | [**ip**] **precedence** *precedence_value* | **mpls experimental** {**topmost**} *experimental _value*}	Mark traffic by setting a new value in the packet, specifying a table map, or specifying a QoS group. • For **cos** *cos_value,* enter a new CoS value to be assigned to the classified traffic. The range is 0 to 7. • For [**ip**] **dscp** *new-dscp*, enter a new DSCP value to be assigned to the classified traffic. The range is 0 to 63 or valid af numbers, cs numbers, **default**, or **ef**. • For [**ip**]**precedence** *new-precedence,* enter a new IP precedence value to be assigned to the classified traffic, 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**). **Note** A class can have either DSCP or precedence marking. If one of these is already configured in a class and you configure the other keyword, the newer command overwrites the previous command. If the value configured for a **set ip precedence** class in an earlier class in a policy overlaps the value configured for a **set ip dscp** class in a later class, the earlier configuration is always matched. • For **mpls experimental topmost** *exp-number*, enter the new MPLS experimental value for the outermost or topmost label. This keyword marks all the valid MPLS packets. The range is 0 to 7. If the policy is already attached to an interface, you must exit the policy-map class configuration mode before the modification is applied.
Step 5	**end**	Return to the privileged EXEC mode.

	Command	Purpose
Step 6	**show policy-map** [*policy-map-name* [**class** *class-map-name*]]	Verify your entries.
Step 7	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

After you create the input policy map, attach it to a target. See the "Attaching a Service Policy to an Interface or EFP" section on page 33-74. Use the **no** form of the appropriate command to delete a policy map or remove an assigned CoS, DSCP, MPLS, or precedence value.

This example uses a policy map to re-mark a packet. The first marking (the **set** command) applies to the QoS default class map that matches all the traffic not matched by class *AF31-AF33* and sets all the 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 gigabitethernet 0/1
Switch(config-if)# service-policy input Example
Switch(config-if)# exit
```

Configuring Policing

Policing is used to enforce a traffic-metering profile. A policer meters a particular traffic flow and determines if a packet in the given flow conforms to the specified rate. 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.

The switch supports 1-rate policing with a 2-color marker. Mapped packets can be sent without modification, dropped, or marked to the options specified by the **set** command. Note that traffic rates are configured in bits per second and burst size is entered in bytes.

Follow these guidelines when configuring policing:

- Hierarchical policing is not supported.

Configuring a Policy Map with 1-Rate, 2-Color Policing

In the privileged EXEC mode, perform these steps to create a 1-rate, 2 color output policy map with individual policing.

	Command	Purpose
Step 1	**configure terminal**	Enter the global configuration mode.
Step 2	**class** {*class-map-name* | **class-default**}	Enter a class-map name or **class-default** to match all unclassified packets, and enter policy-map class configuration mode. If you enter a class-map name, you must have already created the class map by using the class-map global configuration command.

	Command	Purpose
Step 3	**policy-map** *policy-map-name*	Create a policy map by entering the policy map name, and enter policy-map configuration mode. By default, no policy maps are defined.
Step 4	**police** {*rate-bps* | **cir** {*cir-bps* [*burst-bytes*] [**bc-***burst-bytes*]| **percent** *percent* [*burst-ms*] [**bc-***burst-ms*]} }	Configure a traffic policer based on the traffic rate or committed information rate (CIR). By default, no policer is defined. • For *rate-bps,* specify average traffic rate in bits per second (b/s). The range is 64000 to 10000000000. • For **cir** *cir-bps,* specify a committed information rate (CIR) in bits per second (b/s). The range is 64000 to 10000000000. • For *burst-bytes* (optional), specify the normal burst size in bytes. The range is 8000 to 16000000. • For **bc-***burst-bytes* (optional), specify the conformed burst (bc) or the number of acceptable burst bytes. The range is 8000 to 16000000. • For **cir percent** *percent*, specify the rate as a percentage of the bandwidth assigned to the class. The range is 1 to 100 percent. • For *burst-ms* (optional), enter the conform burst size in milliseconds. The range is 1 to 2000. The default is 250 ms. • For **bc-***burst-ms* (optional), specify the conformed burst (bc) in milliseconds. The range is 1 to 2000. **Note** If you are configuring a single action for conformed and exceeded packets, you can specify them in the same line as the **police** command. If configuring multiple actions, press ENTER after the **police** command, and enter policy-map class police configuration mode (config-pmap-c-police) mode to specify the actions to take.
Step 5	**conform-action** {**drop** | **set-cos-transmit** *cos_value* | **set-discard-class** *discard_value* | **set-dscp-transmit** *dscp_value* | **set-mpls-exp-topmost-transmit-***new-exp* | **set-prec-transmit** *value* | **transmit**}	(Optional) Enter the action to be taken on packets that conform to the CIR. • **drop**—Drop the packet. • **set-cos-transmit** *cos_value*—Set the CoS value to a new value and send the packet. The range is 0 to 7. • **set-dscp-transmit** *dscp_value*—Set the IP DSCP value to a new value and send the packet. The range is 0 to 63. You also can enter a mnemonic name for a commonly used value or use the question mark (?) to see a list of available values. • **set-mpls-exp-topmost-transmit** *new-exp*—Enter the new MPLS experimental value for the outermost or topmost label, and send the packet. The range is 0 to 7. • **set-prec-transmit** *value*—Set the IP precedence value to a new value and send the packet. The range is 0 to 7. • **transmit**—Send the packet without altering it. This is the default. **Note** If you are configuring a single action for conforming and exceeding packets, you can specify them in the same line. If configuring multiple actions, press ENTER after the **conform-action** command.

	Command	Purpose
Step 6	**exceed-action** {**drop** | **set-cos-transmit** *cos_value* | **set-discard-class-transmit** *discard_value* | **set-mpls-exp-topmost-transmit** *new-exp* | **set-prec-transmit** *value* | **transmit**}	(Optional) Enter the action to be taken on packets that exceed the CIR. The default exceed action, if no action is configured, is **drop**. • **drop**—Drop the packet. • **set-cos-transmit** *cos_value*—Set the CoS value to a new value, and send the packet. The range is 0 to 7. • **set-discard-class-transmit** *discard_value*—Set the discard value to a new value, and send the packet. The range is 0 to 7. • **set-mpls-exp-topmost-transmit** *new-exp*—Enter the new MPLS experimental value for the outermost or topmost label, and send the packet. The range is 0 to 7. • **set-prec-transmit** *value*—Set the IP precedence value to a new value, and send the packet. The range is 0 to 7. • **transmit**—Send the packet without altering it. **Note** If you explicitly configure **exceed-action drop** as keywords in the command, you must enter the policy-map class police configuration mode and enter the **no exceed-action drop** command to remove the previously configured exceed action before you can enter the new exceed action.
Step 7	**end**	Return to the privileged EXEC mode.
Step 8	**show policy-map** [*policy-map-name* [**class** *class-map-name*]]	Verify your entries.
Step 9	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

After you create the policy map, attach it to an interface or an EFP. See the "Attaching a Service Policy to an Interface or EFP" section on page 33-74.

This example shows how to create a traffic classification with a CoS value of 4, create a policy map, and attach it to an egress port. The average traffic rate is limited to 10000000 b/s with a burst size of 10000 bytes.

```
Switch(config)# class-map video-class
Switch(config-cmap)# match cos 4
Switch(config-cmap)# exit
Switch(config)# policy-map video-policy
Switch(config-pmap)# class video-class
Switch(config-pmap-c)# priority
Switch(config-pmap-c)# police cir 10000000 bc 10000
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# service-policy output video-policy
Switch(config-if)# exit
```

This example shows how to create a policy map with a conform action of **set dscp** and a default exceed action, and attach it to an EFP.

```
Switch(config)# class-map in-class-1
Switch(config-cmap)# match dscp 14
Switch(config-cmap)# exit
Switch(config)# policy-map out-policy
Switch(config-pmap)# class out-class-1
```

```
Switch(config-pmap-c)# priority
Switch(config-pmap-c)# police cir 230000 bc 8000 conform-action set-dscp-transmit 33
exceed-action drop
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet0/1
Switch (config-if)# service instance 1 Ethernet
Switch (config-if-srv)# service-policy output out-policy
Switch (config-if-srv)# exit
```

This example shows how to configure port shaping when EFP policies are present, by configuring a hierarchical policy map that shapes a port to 50 percent, allocated according to the *out-policy* policy map configured in the previous example.

```
Switch(config)# policy-map port-policy
Switch(config-pmap)# class class-default
Switch(config-pmap-c)# shape average percent 50
Switch(config-pmap-c)# service-policy out-policy
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# service-policy output out-policy-parent
Switch(config-if)# exit
```

This example shows how to use the policy-map class police configuration mode to set multiple conform actions and an exceed action. The policy map sets a committed information rate of 23000 bits per second (b/s) and a conform burst size of 10000 bytes. The policy map includes multiple conform actions (for DSCP and Layer 2 CoS) and an exceed action.

```
Switch(config)# class-map cos-set-1
Switch(config-cmap)# match cos 3
Switch(config-cmap)# exit
Switch(config)# policy-map map1
Switch(config-pmap)# class cos-set-1
Switch(config-pmap-c)# priority
Switch(config-pmap-c)# police cir 23000 bc 10000
Switch(config-pmap-c-police)# conform-action set-dscp-transmit 48
Switch(config-pmap-c-police)# conform-action set-cos-transmit 5
Switch(config-pmap-c-police)# exceed-action drop
Switch(config-pmap-c-police)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# service-policy output map1
Switch(config-if)# exit
```

Configuring Class-Based Priority Queuing

You can use the **priority** policy-map class configuration command to ensure that a particular class of traffic is given preferential treatment. Strict priority queuing provides low-latency service to the class.

- When priority is configured in an output policy map without the police command, you can only configure the other queues for sharing by using the bandwidth remaining percent policy-map command to allocate excess bandwidth.
- You can apply priority at the class level and to the VLAN level.
- You can associate the priority command with a single class at the class level and a single class at the VLAN level.

Beginning in privileged EXEC mode, follow these steps to configure a strict priority queue:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**class-map** *class-map-name*	Create classes for three egress queues. Enter match conditions classification for each class.
Step 3	**policy-map** *policy-map-name*	Create a policy map by entering the policy map name, and enter policy-map configuration mode.
Step 4	**class-***class-map-name*	Enter the name of the priority class (created by using the **class-map** global configuration command), and enter policy-map class configuration mode for the priority class.
Step 5	**priority**	Set the strict scheduling priority for this class. **Note** Only one unique class map in an attached policy map can be associated with a **priority** command. You cannot configure priority along with any other queuing action (**bandwidth** or **shape average**).
Step 6	**exit**	Exit policy-map class configuration mode for the priority class.
Step 7	**class** *class-map-name*	Enter the name of a nonpriority class, and enter policy-map class configuration mode for that class.
Step 8	**bandwidth remaining percent** *value*	Set output bandwidth limits for the policy-map class as a percentage of the remaining bandwidth. The range is 0 to 100 percent.
Step 9	**end**	Return to privileged EXEC mode.
Step 10	**show policy-map**	Verify your entries.
Step 11	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

After you have created an output policy map, you attach it to an egress port or EFP. See the "Attaching a Service Policy to an Interface or EFP" section on page 33-74.

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 gigabitethernet0/1
Switch(config-if)# service-policy output policy1
Switch(config-if)# exit
```

Configuring Weighted Tail Drop

Weighted tail drop (WTD) adjusts the queue size associated with a traffic class in terms of time and bytes. You configure WTD by using the **queue-limit** policy-map class configuration command. The queue-limit command is allowed only after you have configured a scheduling action (**bandwidth**, **shape average**, or **priority**).

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

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**policy-map** *policy-map-name*	Create a policy map by entering the policy map name, and enter policy-map configuration mode.
Step 3	**class-**{*class-map-name* | **class-default**}	Enter a child class-map name or enter **class-default** to match all unclassified packets, and enter policy-map class configuration mode. • If you enter a class-map name, you must perform Step 4 to configure a scheduling action (**bandwidth**, **shape average**, or **priority**) before you go to Step 5 to configure queue-limit. • If you enter **class-default**, you can omit Step 4.
Step 4	**bandwidth** {*rate* | **percent** *value* | **remaining percent** *value*} or **shape average** {*target bps* | **percent** *value*} or **priority**	Configure a scheduling action for the traffic class.

	Command	Purpose
Step 5	**queue-limit** [**cos** *value* | **discard-class** *value* | **dscp** *value* | **exp** *value* | **precedence** *value* | **qos-group** *value* | **percent** *value*] *number-of-packets* [**packets**] | *limit* [**bytes** | **us**]]	Specify the queue size for the traffic class. • (Optional) For **cos** *value*, specify a CoS value. The range is from 0 to 7. • (Optional) Enter **discard-class** *value* to specify the drop precedence for a packet during congestion management. The range is 0 to 7. • (Optional) For **dscp** *value*, specify a DSCP value. The range is from 0 to 63. • (Optional) For **exp** *value*, specify an MPLS experimental value. The range is from 0 to 7. • (Optional) For **precedence** *value*, specify an IP precedence value. The range is from 0 to 7. • (Optional) For **qos-group** *value*, enter a QoS group value. The range is from 0 to 99. • (Optional) For **percent** *value*, enter a precentage value of the threshold. • 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. The value is specified in packets by default, but the **packets** keyword is optional. • For **bytes** *bytes*, enter the maximum threshold in bytes. The range is from 200 to 2097152. The default depends on the interface speed. On 10/100/1000 Mb/s interfaces, the default is 12000 (12 K) bytes. On 10 Gb/s interfaces, the default is 12 0000 (120 K) bytes. • For **us** *microseconds*, enter the maximum threshold in microseconds. This is the default for specifying threshold. The range is from 1 to 1778. The default depends on the interface speed. On 10 Mb/s interfaces, the default is 10000 us. On 100 Mb/s interfaces, the default is 1000 us. On 1000 Mb/s and 10 Gb/s interfaces, the default is 100 us. • If you do not enter **bytes** *bytes* or **us** *microseconds*, the default is **us**.
Step 6	**end**	Return to privileged EXEC mode.
Step 7	**show policy-map** [*policy-map-name* [**class** *class-map-name*]]	Verify your entries.
Step 8	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

After you have created an output policy map, you attach it to an egress port. See the "Attaching a Service Policy to an Interface or EFP" section on page 33-74.

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

Note If the device connected to a 10/100/1000 Mb/s interface starts bursting the traffic more than the 1 Gigabit line rate because of multicast replication, you should increase the queue-limit for the policy map class applied to the interface to 48000 (48 K) bytes because the interface cannot handle the excess burst.

This example shows a policy map with a specified bandwidth and queue size. Traffic that is not DSCP 30 or 10 is assigned a queue-limit of 2000 bytes. Traffic with a DSCP value of 30 is assigned a queue-limit of 1000 bytes, and traffic with a DSCP value of 10 is assigned a queue-limit of 1500 bytes. 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 3000 bytes.

```
Switch(config)# policy-map gold-policy
Switch(config-pmap)# class traffic
Switch(config-pmap-c)# bandwidth percent 50
Switch(config-pmap-c)# queue-limit bytes 2000
Switch(config-pmap-c)# queue-limit dscp 30 bytes 1000
Switch(config-pmap-c)# queue-limit dscp 10 bytes 1500
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 bytes 3000
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# service-policy output gold-policy
Switch(config-if)# exit
```

There can be only three unique qualified queue-limit thresholds. In this example, there are four unique thresholds, so the configuration is rejected:

```
Switch(config-pmap-c)# queue-limit 100 us
Switch(config-pmap-c)# queue-limit cos 2 200 us
Switch(config-pmap-c)# queue-limit cos 3 300 us
Switch(config-pmap-c)# queue-limit cos 4 400 us
```

In the next example, although there appear to be only three unique thresholds, in reality there are four threshold configurations, including an implied default threshold. The configuration is rejected.

```
Switch(config-pmap-c)# queue-limit cos 2 200 us
Switch(config-pmap-c)# queue-limit cos 3 300 us
Switch(config-pmap-c)# queue-limit cos 4 400 us
```

In this last example, there are only three unique thresholds and the configuration is allowed.

```
Switch(config-pmap-c)# queue-limit 100 us
Switch(config-pmap-c)# queue-limit cos 2 100 us
Switch(config-pmap-c)# queue-limit cos 3 300 us
Switch(config-pmap-c)# queue-limit cos 4 400 us
```

Configuring Weighted Random Early Detection

This section describes the tasks involved in configuring Weighted Random Early Detection (WRED).

Restrictions and Usage Guidelines

- Whales supports the following commands:

  - WRED based on Outer COS (random-detect cos-based)
  - WRED based on IPv4 DSCP (random-detect dscp-based)
  - WRED based on IPv4 Precedence (random-detect precedence-based)
  - WRED based on discard class (random-detect discard-class-based)
  - WRED based on EXP
- WRED polices is supported on below targets.
  - L3 Main interface
  - Switchport interface
  - Service instance
  - Port-channel Member-link interface.
- Maximum of 2 WRED curves are supported.
- WRED configuration is supported only in egress policy-map.
- Default value for exponential-weighting-constant is 9.
- Default value for mark-probability is 10.
- Minimum-threshold and maximum-threshold can be specified in terms of packets only.
- WRED command is supported either with shape or CBWFQ command.
- WRED is supported in PHB level. Not supported in logical/physical level classes
- MQC based WRED counters are not supported.
- WRED Non-aggregate mode is not supported.
- WRED is not supported in priority queues.

When a packet arrives, the following events occur:

1. The average queue size is calculated.
2. If the average is less than the minimum queue threshold, the arriving packet is queued.
3. If the average is between the minimum queue threshold for that type of traffic and the maximum threshold for the interface, the packet is either dropped or queued, depending on the packet drop probability for that type of traffic.
4. If the average queue size is greater than the maximum threshold, the packet is dropped.

See Weighted Random Early Detection (WRED), page 33-22 for more details on how WRED works.

To configure WRED on a switch, perform the tasks described in the following sections. The tasks described in the first section are required; the tasks described in the remaining sections are optional.

- Enabling WRED, page 33-68 (Required)
- Changing the WRED Parameters, page 33-68 (Optional)

Note WTD and WRED cannot be configured under the same class of a policy map.

Enabling WRED

In the privileged EXEC mode, perform the following steps to enable WRED for a class in a policy map.

	Command	Purpose
Step 1	**configure terminal**	Enable the global configuration mode.
Step 2	**policy-map** *policy-map-name*	Create a policy map by entering the policy map name, and enter policy-map configuration mode.
Step 3	**class**-{*class-map-name*| **class-default**}	Enter a child class map name or enter the **class-default** command to match all the unclassified packets, and enter the policy-map class configuration mode.
Step 4	**random-detect** [**dscp-based** | **prec-based**]	Enables WRED.
Step 5	**random-detect precedence** *precedence min-threshold max-threshold mark-prob-denominator*	Configures parameters for packets with a specific IP precedence. The minimum threshold for IP precedence 0 corresponds to half the maximum threshold for the interface. Repeat this command for each precedence. To configure RED, rather than WRED, use the same parameters for each precedence. • For **bytes** *bytes*, enter the maximum threshold in bytes. The default depends on the interface speed. The minimum and maximum range is 1 to 4096.
Step 6	**end**	Return to the privilege EXEC mode.
Step 7	**show policy-map** [*policy-map-name* [**class** *class-map-name*]]	Verify your entries.
Step 8	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

The following example configures a policy for a class called acl10 included in a policy map called policy10:

```
Switch# configure terminal
Switch(config)# policy-map policy10
Switch(config-pmap)# class acl10
Switch(config-pmap-c)# random-detect dscp-based
Switch(config-pmap-c)# random-detect precedence 3 500 1000 10
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
```

Changing the WRED Parameters

To change the WRED parameters, use the following commands in the interface configuration mode, as needed.

	Command	Purpose
Step 1	**configure terminal**	Enable global configuration mode.
Step 2	**policy-map** *policy-map-name*	Creates a policy map by entering the policy map name, and enter the policy-map configuration mode.
Step 3	**class**-{*class-map-name*| **class-default**}	Enter a child class map name or enter the **class-default** command to match all the unclassified packets, and enter the policy-map class configuration mode.

	Command	Purpose
Step 4	**random-detect exponential-weighting-constant** *exponent*	Configures the weight factor used in calculating the average queue length.
Step 5	**random-detect precedence** *precedence min-threshold max-threshold mark-prob-denominator*	Configures parameters for packets with a specific IP precedence. The minimum threshold for IP precedence 0 corresponds to half the maximum threshold for the interface. Repeat this command for each precedence. To configure RED rather than WRED, use the same parameters for each precedence. • For **bytes** *bytes*, enter the maximum threshold in bytes. The default depends on the interface speed. The minimum and maximum range is 1 to 4096.
Step 6	**random-detect cos** *cos-value min-threshold max-threshold mark-probability-denominator*	Configures parameters for packets with a specific outer class of service (CoS) value. • For **bytes** *bytes*, enter the maximum threshold in bytes. The default depends on the interface speed. The minimum and maximum range is 1 to 4096.
Step 7	**random-detect dscp** *dscp-value min-threshold max-threshold* [*mark-probability-denominator*]	Configures the minimum and maximum packet thresholds for the differentiated services code point (DSCP) value. • For **bytes** *bytes*, enter the maximum threshold in bytes. The default depends on the interface speed. The minimum and maximum range is 1 to 4096.
Step 8	**end**	Return to the privileged EXEC mode.
Step 9	**show policy-map** [*policy-map-name* [**class** *class-map-name*]]	Verify your entries.
Step 10	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

When you enable WRED with the **random-detect** interface configuration command, the parameters are set to their default values. The weight factor is 9. For all precedences, the mark probability denominator is 10, and the maximum threshold is based on the output buffering capacity and the transmission speed for the interface.

The default minimum threshold depends on the precedence. The minimum threshold for IP precedence 0 corresponds to half the maximum threshold. The values for the remaining precedences fall between half the maximum threshold and the maximum threshold at evenly spaced intervals.

Hierarchical Policy Maps Configuration Examples

These are examples of using policy maps to configure hierarchical QoS.

Configure policy maps *phb* and *vlan*:

```
Switch(config)# policy-map phb
Switch(config-pmap)# class cos1
Switch(config-pmap-c)# shape average 1000
Switch(config-pmap-c)# exit
Switch(config-pmap)# class cos2
Switch(config-pmap-c)# shape average 2000
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
```

```
Switch(config)# policy-map vlan
Switch(config-pmap)# class vlan1
Switch(config-pmap-c)# shape average 5000
Switch(config-pmap-c)# service-policy phb
Switch(config-pmap-c)# exit
Switch(config-pmap)# class vlan2
Switch(config-pmap-c)# shape average 6000
Switch(config-pmap-c)# service-policy phb
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
```

This is an example of a policy on a port to address Port-Shaper when EFP policies are present:

```
Switch(config)# policy-map port-policy
Switch(config-pmap)# class class-default
Switch(config-pmap-c)# shape average percent 50
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config-pmap)# policy-map efp-policy
Switch(config-pmap)# class class-default
Switch(config-pmap-c)# shape average percent 25
Switch(config-pmap-c)# service-policy child-policy
Switch(config-pmap-c)# exit
```

This is an example of a 3-level output policy. You can attach this policy only to physical ports and not to EFP service instances.

```
Switch(config)# policy-map interface-policy-with-vlan-child
Switch(config-pmap)# class class-default
Switch(config-pmap-c)# shape average 10000
Switch(config-pmap-c)# service-policy vlan-policy
Switch(config-pmap-c)# exit
```

This is an example of a 2-level output policy. You can attach this policy to physical ports or to EFP service instances.

```
Switch(config)# policy-map interface-policy-with-phb-child
Switch(config-pmap)# class class-default
Switch(config-pmap-c)# shape average 10000
Switch(config-pmap-c)# service-policy phb
Switch(config-pmap-c)# exit
```

This is an example of a 2-level output policy. You can attach this policy to physical ports or to EFP service instances.

```
Switch(config)# policy-map interface-policy-with-phb-child
Switch(config-pmap)# class class-default
Switch(config-pmap-c)# service-policy vlan-policy
Switch(config-pmap-c)# exit
```

Configuring Table Maps

You can configure table maps to manage a large number of traffic flows with a single command. You use table maps to correlate specific DSCP, IP precedence and CoS values to each other, to mark down a DSCP, IP precedence, or CoS value, or to assign default values.

These table maps are supported on the switch:

- DSCP to CoS, precedence, or DSCP
- CoS to DSCP, precedence, or CoS

- Precedence to CoS, DSCP, or precedence

Note these guidelines when configuring table maps:

- The switch supports a maximum of 256 unique table maps.
- The maximum number of map statements within a table map is 64.
- Table maps cannot be used in output policy maps.
- Table maps are not supported under class-default class map.
- Dynamic modifications to table maps is not supported. To make changes to the table map you must detach the policy-map from the interface. Make any necessary changes to the policy map and then re-attach it to the interface.

Beginning in privileged EXEC mode, follow these steps to create a table map:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**table-map** *table-map-name*	Create a table map by entering a table-map name and entering table-map configuration mode.
Step 3	**map from** *from-value* **to** *to-value*	Enter the mapping values to be included in the table. For example, if the table map is a DSCP-to-CoS table map, the *from-value* would be the DSCP value and the *to_value* would be the CoS value. Both ranges are from 0 to 63. Enter this command multiple times to include all the values that you want to map.
Step 4	**default** {*default-value* | **copy** | **ignore**}	Set the default behavior for a value not found in the table map. • Enter a *default-value* to specify a certain value. For example, in a DSCP-to-CoS table map, this would be a specific CoS value to apply to all unmapped DSCP values. The range is from 0 to 63. • Enter **copy** to map unmapped values to an equivalent value. In a DSCP-to-CoS table map, this command maps all unmapped DSCP values to the equivalent CoS value. • Enter **ignore** to leave unmapped values unchanged. In a DSCP-to-CoS table map, the switch does not change the CoS value of unmapped DSCP values.
Step 5	**end**	Return to privileged EXEC mode.
Step 6	**show table-map** [*table-map-name*]	Verify your entries.
Step 7	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To delete a table map, use the **no table-map** *table-map-name* global configuration command.

This example shows how to create a DSCP-to-CoS table map. A complete table would typically include additional map statements for the higher DSCP values. The default of 4 in this table means that unmapped DSCP values will be assigned a CoS value of 4.

```
Switch(config)# table-map dscp-to-cos
Switch(config-tablemap)# map from 1 to 1
Switch(config-tablemap)# map from 2 to 1
Switch(config-tablemap)# map from 3 to 1
Switch(config-tablemap)# map from 4 to 2
Switch(config-tablemap)# map from 5 to 2
```

```
Switch(config-tablemap)# map from 6 to 3
Switch(config-tablemap)# default 4
Switch(config-tablemap)# end
Switch# show table-map dscp-to-cos
```

Configuring MPLS and EoMPLS QoS

Default MPLS and EoMPLS QoS Configuration

QoS is disabled. Packets are not modified, and the CoS, DSCP, and IP precedence values in the packet are not changed. Traffic is switched in pass-through mode (packets are switched without any rewrites and classified as best effort without any policing).

The default behavior for VLAN and port-based EoMPLS packets is to use a value of 0 in the EXP bits of the virtual-connection and tunnel labels. The default behavior for L3VPN MPLS packets is to relay the IP Precedence bits into the EXP bits of the virtual-connection and tunnel labels. You can change the default behavior for VLAN- or port-based EoMPLS by applying a hierarchical QoS policy.

MPLS QoS Configuration Guidelines

- The switch supports these MPLS QoS features:
  - MPLS can tunnel the QoS values of a packet (that is, QoS is transparent from edge to edge). With QoS transparency, the IP marking in the IP packet is preserved across the MPLS network.
  - You can mark the MPLS EXP field differently and separately from the per-hop behavior (PHB) marked in the IP precedence, DSCP, or CoS field.
- One label-switched path (LSP) can support up to eight classes of traffic (that is, eight PHBs) because the MPLS EXP field is a 3-bit field.
- MPLS DiffServ tunneling modes support E-LSPs. An E-LSP is an LSP on which nodes determine the QoS treatment for MPLS packet exclusively from the EXP bits in the MPLS header.
- The switch does not support egress QoS marking.
- MPLS QoS classification does not work for bridged MPLS packets.
- For port-based EoMPLS, you cannot match the payload VLAN.

Setting the Priority of Packets with Experimental Bits

MPLS and EoMPLS provide QoS on the ingress router by using 3 experimental bits in a label to determine the priority of packets. To support QoS between LERs, set the experimental bits in both the virtual-connection and tunnel labels.

The process includes these steps on the ingress router:

- Configure a class map to classify IP packets according to their DSCP or IP precedence classification.

- Configure a policy map to mark MPLS packets (write their classification into the MPLS experimental field).
- Attach the service policy to the input interface or service instance.

Beginning in privileged EXEC mode, follow these steps to set the experimental bits for EoMPLS or MPLS QoS:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**class-map** *class-map-name*	Specify the name of the traffic class, and enter class-map configuration mode.
Step 3	**match** {**ip dscp** *dscp-list* | **ip precedence** *ip-precedence-list*}	Specify the matching criteria for IEEE 802.1Q packets. • **ip dscp** *dscp-list*—list of up to eight IP DSCP values to match against incoming packets. The range is 0 to 63. • **ip precedence** *ip-precedence-list*—list of up to eight IP precedence values to match against incoming packets. Separate each value with a space. The range is 0 to 7.
Step 4	**exit**	Return to global configuration mode.
Step 5	**policy-map** *policy-map-name*	Specify the name of the traffic policy to configure, and enter policy-map configuration mode.
Step 6	**class** *class-name*	Specify the name of the predefined traffic class configured with the **class-map** command, and enter policy-map class configuration mode.
Step 7	**set mpls experimental** *exp-number*	Specify the value to which the MPLS bits are set if the packets match the specified policy map. The range is 0 to 7.
Step 8	**exit**	Return to policy-map configuration mode.
Step 9	**exit**	Return to global configuration mode.
Step 10	**interface** *interface-id*	Enter the interface ID, and enter interface configuration mode. The interface should be the ES egress port of the ingress router.
Step 11	**service-policy output** *policy-map-name*	Attach the specified policy map to the output interface.
Step 12	**end**	Return to privileged EXEC mode.
Step 13	**show policy-map** [**policy-map-name** [**class** *class-map-name*]] **show policy-map interface** *interface-id*	Verify the configuration.
Step 14	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To delete an existing policy map, use the **no policy-map** *policy-map-name* global configuration command. To delete an existing class, use the **no class** *class-name* policy-map configuration command.

This example shows how to use class and policy maps to configure different experimental bit settings for DSCP and IP precedence for MPLS QoS.

```
Switch(config)# class-map match-all gold-class
Switch(config-cmap)# match ip dscp 1
Switch(config-cmap)# exit
Switch(config)# class-map match-all silver-class
Switch(config-cmap)# match ip precedence 2
Switch(config-cmap)# exit
```

```
Switch(config)# policy-map in-policy
Switch(config-pmap)# class gold-class
Switch(config-pmap-c)# set mpls experimental 5
Switch(config-pmap-c)# exit
Switch(config-pmap)# class silver-class
Switch(config-pmap-c)# set mpls experimental 4
Switch(config-pmap-c)# exit

Switch(config)# interface gigabitethernet1/1/1
Switch(config-if)# service-policy input in-policy
Switch(config-if)# end
```

MPLS DiffServ Tunneling Modes

The switch supports MPLS DiffServ tunneling modes, which allows QoS to be transparent from one edge of a network to the other edge. A tunnel starts where there is label imposition and ends where there is label disposition.

The switch supports three tunnelling modes:

- uniform mode
- short-pipe mode
- pipe mode

For additional information, see "MPLS DiffServ Tunneling Modes" section of the *MPLS: Traffic Engineering: DiffServ Configuration Guide, Cisco IOS Release 15.x*.

Attaching a Service Policy to an Interface or EFP

You use the **service-policy** interface configuration command to attach a service policy to an interface or EFP (service instance) 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 to a physical port or to an EFP. However, you cannot attach a service policy to a physical port that has EFPs configured. If you try to configure an EFP on a switchport that has a service policy attached, the service policy is detached. However, when EFPs are configured on a physical port, you can attach a port policy with a class-default shaper configuration.

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 or EFP, a warning message that lists the interfaces from which the policy map is being detached is displayed. For example:

```
Warning: Detaching Policy test1 from Interface GigabitEthernet0/1
```

The policy map is then detached and deleted.

Beginning in privileged EXEC mode, follow these steps to attach a policy map to a port:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**interface** *interface-id*	Specify the port to attach to the policy map, and enter interface configuration mode. Valid interfaces are physical ports.

	Command	Purpose
Step 3	**service-policy** {**input** | **output**} *policy-map-name*	Specify the policy-map name and whether it is an input policy map or an output policy map.
Step 4	**end**	Return to privileged EXEC mode.
Step 5	**show policy-map** [*policy-map-name*] [**class** *class-map-name*]	Verify your entries.
Step 6	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To remove the policy map and port association, use the **no service-policy** {**input** | **output**} *policy-map-name* interface configuration command.

Beginning in privileged EXEC mode, follow these steps to attach a policy map to an EFP:

	Command	Purpose
Step 1	**configure terminal**	Enter global configuration mode.
Step 2	**interface** *interface-id*	Specify the port to attach to the policy map, and enter interface configuration mode. Valid interfaces are physical ports.
Step 3	**service instance** *number* **ethernet** [*name*]	Configure an EFP (service instance) and enter service-instance configuration mode. See Chapter 12, "Configuring Ethernet Virtual Connections (EVCs)." **Note** You must enter the **switchport mode trunk** and **switchport trunk allowed vlan none** interface configuration commands on an interface before configuring an EFP. You must also enter the **no ip igmp-snooping vlan** *vlan-id* command for the VLAN of the bridge domain.
Step 4	**encapsulation** {**default** | **dot1q** | **priority-tagged** | **untagged**}	Configure encapsulation type for the service instance. **Note** You must configure encapsulation type and a bridge domain for the service instance, or the **service-policy** command will be rejected.
Step 5	**bridge-domain** *bridge-id* [**split-horizon group** *group-id*]	Configure the bridge domain ID. The range is from 1 to 8000.
Step 6	**service-policy** {**input** | **output**} *policy-map-name*	Specify the policy-map name and whether it is an input policy map or an output policy map.
Step 7	**end**	Return to privileged EXEC mode.
Step 8	**show ethernet service instance policy-map**	Verify your entries.
Step 9	**copy running-config startup-config**	(Optional) Save your entries in the configuration file.

To remove the policy map and port association, use the **no service-policy** {**input** | **output**} *policy-map-name* service-instance configuration command.

Displaying QoS Information

To display QoS information, use one or more of the privileged EXEC commands in Table 33-2. For explanations about available keywords, see the command reference for this release.

Table 33-2 Commands for Displaying Standard QoS Information

Command	Purpose
show class-map [*class-map-name*]	Display QoS class-map information for all class maps or the specified class map.
show policy-map [*policy-map-name* | **interface** [*interface-id*] [**input** | **output**] [**class** *class-name*]]	Display QoS policy map information for the specified policy map name, interface, input or output policy maps, or policy-map class.
show ethernet service-instance policy map	Display QoS policy map information for policy maps attached to EFP service instances.
show running-config	Display the configured class maps, policy maps, table maps, and aggregate policers.

You can use the **show policy-map interface** [*interface-id*] privileged EXEC command to show input and output policy map statistics per classification. Statistics include the number of packets that match each specified traffic stream, the corresponding configured action, such as policing or scheduling, and the associated statistics.

This is an example of the output of the **show policy-map interface** command showing statistics for an output policy map.

```
Switch# show policy-map interface gigabitethernet 0/2
 GigabitEthernet0/2

  Service-policy output: phb

    Class-map: phb (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: cos  2
          Bandwidth 1000 (kbps)
          Queue-limit current-queue-depth 0 bytes
              Output Queue:
                Tail Packets Drop: 0
                Tail Bytes Drop: 0

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