Quality of Service Configuration

This chapter describes how to configure Quality of Service (QoS) by using the modular QoS Command Line Interface (CLI), or MQC, commands on the CGR 2010 ESM. With QoS, you can provide preferential treatment to certain types of traffic at the expense of others. When QoS is not configured, the switch module offers best-effort service to each packet, regardless of the packet contents or size. It sends the packets without any assurance of reliability, delay bounds, or throughput. MQC provides a comprehensive hierarchical configuration framework for prioritizing or limiting specific streams of traffic.

This chapter also describes QoS features for implementing high-priority (low-latency) traffic via the internal data path between the CGR 2010 ESM and the host Cisco CGR 2010 router. The internal data path is called *PortChannel48 (see "Implementing High-Priority Traffic to the Host Router" section on page 14-95)*. For more information, see also Chapter 9, "EtherChannel Configuration Between the Switch Module and the Host Router."

Note IPv6 QoS is not supported.

For more information about Cisco IOS MQC commands, see the "Cisco IOS Quality of Service Solutions Command Reference" at this site:

http://www.cisco.com/en/US/docs/ios/12_2/qos/command/reference/fqos_r.html

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

For information about using Ethernet terminal loopback to test full-path QoS on an interface, see the "Enabling Ethernet Loopback" section in Chapter 45, "Configuring Ethernet OAM, CFM, and E-LMI" in the *Cisco CGS 2520 Software Configuration Guide*.

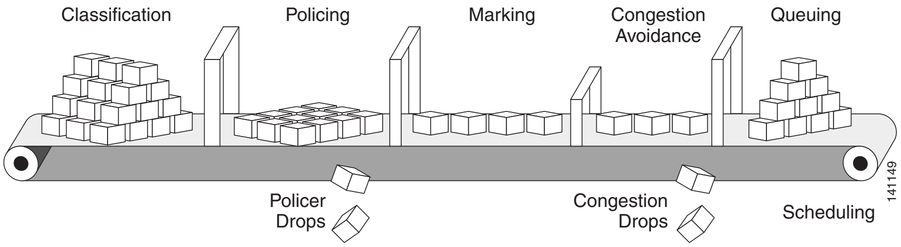
Understanding QoS

Typically, networks operate on a best-effort delivery basis, which means that 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 the QoS feature, 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 14-1 shows the Modular QoS CLI (MQC) model.

Figure 14-1 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 module identifies all key packet fields: class of service (CoS), Differentiated Services Code Point (DSCP), or IP precedence. The switch module classifies the packet based on this content or based on an access-control list lookup. For more information, see the "Classification" section on page 14-6.
- *Packet policing* determines whether a packet is in or out of profile by comparing the rate of the incoming traffic to the configured policer. 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 control the traffic flow for packets that conform to or exceed 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). For more information, see the "Policing" section on page 14-15.
- *Packet prioritization or marking* evaluates the classification and policer information to determine the action to take. 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 reclassified. For more information, see the "Marking" section on page 14-21.
- *Congestion management* uses queuing and scheduling algorithms to queue and sort traffic that is leaving a port. The switch module 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. For more information, see the "Congestion Management and Scheduling" section on page 14-26.
- *Weighted tail-drop (WTD).* Queuing on the switch module is enhanced with the weighted tail-drop (WTD) algorithm, a congestion-avoidance mechanism. WTD differentiates traffic classes and regulates the queue size (in number of packets) based on the classification. For more information, see the "Congestion Avoidance and Queuing" section on page 14-32.

This section includes information about these topics:

- Modular QoS CLI, page 14-3
- Input and Output Policies, page 14-4
- Classification, page 14-6
- Table Maps, page 14-14
- Policing, page 14-15
- Marking, page 14-21
- Congestion Management and Scheduling, page 14-26
- Congestion Avoidance and Queuing, page 14-32

Modular QoS CLI

Modular QoS CLI (MQC) allows users to create traffic policies and attach these policies to interfaces. A traffic policy contains a traffic class and one or more QoS features. You use a traffic class to classify traffic, and the QoS features in the traffic policy determine how to treat the classified traffic.

Modular QoS CLI configuration includes these steps:

Step 1 Define a traffic class.

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

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

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

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

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

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

- You name the traffic policy in the **policy-map** command line to enter policy-map configuration mode.

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

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

Step 3 Attach the traffic policy to an interface.

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

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

Input and Output Policies

Policy maps are either input policy maps or output policy maps, attached to packets as they enter or leave the switch module by service policies applied to interfaces. Input policy maps perform policing and marking on received traffic. Policed packets can be dropped or reduced in priority (marked down) if they exceed the maximum permitted rates. Output policy maps perform scheduling and queuing on traffic as it leaves the switch module.

Input policies and output policies have the same basic structure; the difference is in the characteristics that they regulate. Figure 14-2 shows the relationship of input and output policies.

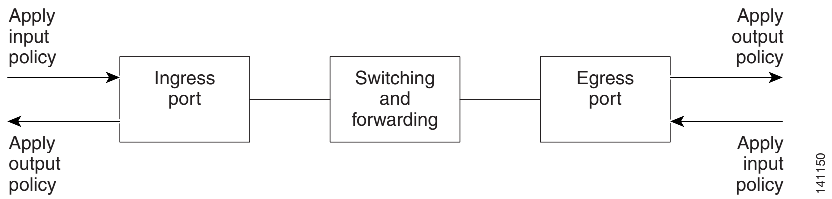
The number of configurable policer profiles on the switch module is 256. The number of supported policer instances on the switch module is 1,024 minus 1 more than the number of interfaces on the switch module.

The number of policer instances for the Copper model (GRWIC-D-ES-2S-8PC), which has ten interfaces, is 1,013; the number of policer instances for the SFP model (GRWIC-D-ES-6S), which has six interfaces, is 1,017.

- You can use a policer profile in multiple instances
- You can configure a maximum of 256 policy maps
- You can apply one input policy map and one output policy map to an interface.

When CPU protection is enabled (the default), you can configure 45 ingress policers per port. If you disable CPU protection by entering the **no policer cpu uni all** global configuration command and reloading the switch module, you can configure a maximum of 63 policers per port for user-defined classes and one for class-default. You can enter the **show policer cpu uni-eni** {**drop** | **rate**} privileged EXEC command to see if CPU protection is enabled.

Figure 14-2 Input and Output Policy Relationship



Input Policy Maps

Input policy map classification criteria include matching a CoS, a DSCP, or an IP precedence value or matching an access control list (ACL) or VLAN ID (for per-port, per-VLAN QoS). Input policy maps can have any of these actions:

- Setting or marking a CoS, a DSCP, an IP precedence, or QoS group value
- Individual policing
- Aggregate policing

Only input policies provide matching on access groups or VLAN IDs, and only output policies provide matching on QoS groups. You can assign a QoS group number in an input policy and match it in the output policy. The class **class-default** is used in a policy map for any traffic that does not explicitly match any other class in the policy map. Input policy maps do not support queuing and scheduling keywords, such as **bandwidth**, **queue-limit**, **priority**, and **shape average**.

An input policy map can have a maximum of 64 classes plus **class-default**. You can configure a maximum of 64 classes in an input policy.

Output Policy Maps

Output policy map classification criteria include matching a CoS, a DSCP, an IP precedence, or a QoS group value. Output policy maps can have any of these actions:

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

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 14-12 for more information.

Output policies do not support marking or policing (except in the case of priority with policing). There is no egress packet marking on the switch module (no **set** command in an output policy).

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

An output policy map attached to an egress port can match only the packets that have already been matched by an input policy map attached to the ingress port for the packets. You can attach an output policy map to any or all ports on the switch module. The switch module supports configuration and attachment of a unique output policy map for each port. 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 ports on the switch module. There are no limitations on the configurations of bandwidth, priority, or shaping.

You can configure the output policy classification criteria for CPU-generated traffic by using the **cpu traffic qos** [**cos** *value* | **dscp** *value* | **precedence** *value* | **qos-group** *value*] global configuration command.

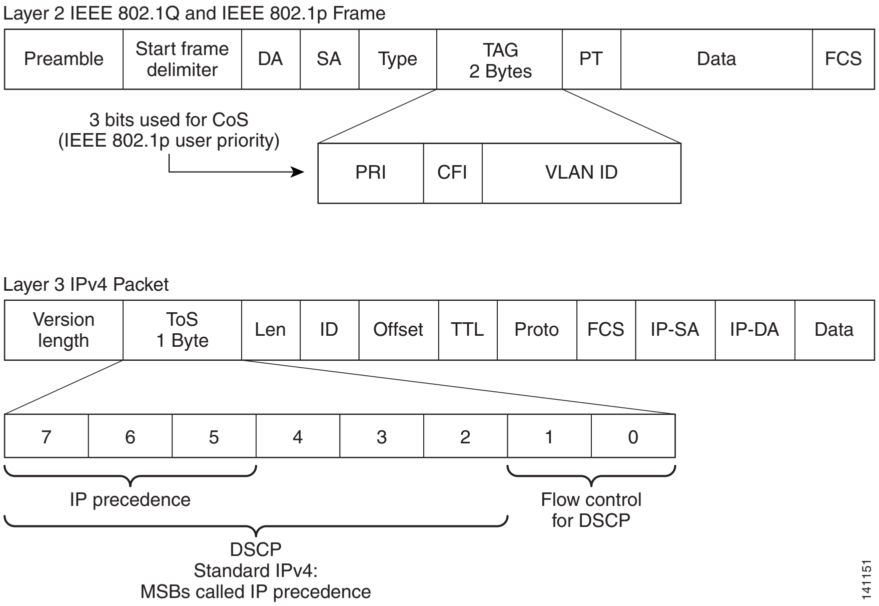
Classification

Classification distinguishes one kind of traffic from another by examining the fields in the packet header. When a packet is received, the switch module examines the header and identifies all key packet fields. A packet can be classified based on an ACL, on the DSCP, the CoS, or the IP precedence value in the packet, or by the VLAN ID. Figure 14-3 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.

- On ports configured as Layer 2 IEEE 802.1Q trunks, all traffic is in 802.1Q frames except for traffic in the native VLAN. Layer 2 802.1Q frame headers have a 2-byte Tag Control Information field that carries the CoS value, called the User Priority bits, in the three most-significant bits, and the VLAN ID value in the 12 least-significant bits. Other frame types cannot carry Layer 2 CoS values.
- Layer 2 CoS values range from 0 to 7.
- 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.
- Output remarking is based on the Layer 2 or Layer 3 marking type, marking value and packet type.

Figure 14-3 QoS Classification Layers in Frames and Packets



These sections contain additional information about classification:

- "Class Maps" section on page 14-7
- "The match Command" section on page 14-7
- "Classification Based on Layer 2 CoS" section on page 14-8
- "Classification Based on IP Precedence" section on page 14-8
- "Classification Based on IP DSCP" section on page 14-8
- "Classification Comparisons" section on page 14-10
- "Classification Based on QoS ACLs" section on page 14-11
- "Classification Based on QoS Groups" section on page 14-12
- "Classification Based on VLAN IDs" section on page 14-13

Class Maps

As explained previously, you use an MQC class map to name a specific traffic flow (or class) and to isolate it from all other traffic. A class map defines the criteria used to match against a specific traffic flow to further classify it. If you have more than one type of traffic that you want to classify, you can create another class map and use a different name. When you enter the **class-map** command with a class-map name, the switch module enters the class-map configuration mode. In this mode, you define the match criterion for the traffic by using the **match** class-map configuration command. After a packet is matched against the class-map criteria, it is acted on by the associated action specified in a policy map.

You can match more than one criterion for classification. You can 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 to enter class map configuration mode.

Note 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.

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

The *match* Command

To configure the type of content used to classify packets, you use the **match** class-map configuration command to specify the classification criteria. If a packet matches the configured criteria, 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, and IP precedence values. These values are referred to as *markings* on a packet. You can also match an access group, a QoS group, or a VLAN ID or ID range for per-port, per-VLAN QoS.

- For an input policy map, you cannot configure an IP classification (**match ip dscp**, **match ip precedence**, **match access-group** for an IP ACL) and a non-IP classification (**match cos** or **match access-group** for a MAC ACL) in the same policy map or class map.

- When an input policy map with only Layer 2 classification is attached to a routed port or a switch module port containing a routed switch module virtual interface (SVI), the service policy acts only on switching eligible traffic and not on routing eligible traffic.
- On an 802.1Q tunnel port, you can use only an input policy map with Layer 2 classification based on MAC ACLs to classify traffic. Input policy maps with Layer 3 classification, match Layer 2 CoS classification, or per-port, per-VLAN policies are not supported on tunnel ports.
- In an output policy map, no two class maps can have the same classification criteria, that is, the same match qualifiers and values.

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
```

Classification Based on Layer 2 CoS

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

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

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 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* provides delivery of 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 provides the best probability of a packet being forwarded from one end of the network to the other.
- Entering Class Selector (CS) service values of 1 to 7, corresponding to IP precedence bits in the ToS field of the packet.
- Using Expedited Forwarding (EF) to specify a low-latency path. This corresponds to a DSCP value of 46. EF services use priority queuing to preempt lower priority traffic classes.

This display shows the available classification options:

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

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

802.1Q Tunneling CoS Mapping

The switch module supports VLAN mapping from the customer VLAN-ID (C-VLAN) to a service-provider VLAN-ID (S-VLAN). See the

“Understanding VLANs” section on page 11-1. For QoS, the switch module can set the service-provider CoS (S-CoS) from either the customer CoS (C-CoS) or the customer DSCP (C-DSCP) value, and can map the inner CoS to the outer CoS for any traffic with traditional 802.1Q tunneling (QinQ) or selective QinQ VLAN mapping. This default allows copying the customer CoS into the service provider network.

The switch module supports C-CoS to S-CoS propagation for traditional QinQ and for selective QinQ on trunk ports. This is the default behavior and does not require configuration. When you configure traditional QinQ or selective QinQ on Layer 2 trunk ports using 1-to-2 VLAN mapping, the switch module also allows setting the S-CoS from C-DSCP.

For traffic entering the switch module on 802.1Q tunnel ports or trunk ports configured for VLAN mapping, the switch module has the ability to examine the customer packet header and set the service-provider CoS value (S-CoS) from either the customer CoS value or the customer DSCP value.

Configuring CoS matching on 802.1Q mapped ports is handled in this way:

- On interfaces configured for 802.1Q tunneling (on tunnel or trunk ports) or selective 802.1Q (on trunk ports), 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.
- 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 and selective QinQ mapping interfaces, a DSCP match sets the outer CoS of the encapsulated value.
- You can set DSCP based on matching the outer VLAN.
- If you enter the **match cos** command on interfaces configured for traditional QinQ or for selective QinQ mapping, the match is to the outer CoS, which is the reflected inner Cos (C-CoS).

Classification Comparisons

Table 14-1 shows suggested IP DSCP, IP precedence, and CoS values for typical traffic types.

Table 14-1 Typical Traffic Classifications

Traffic Type	DSCP per-hop	DSCP (decimal)	IP Precedence	CoS
Voice-bearer—traffic in a priority queue or the queue with the highest service weight and lowest drop priority.	EF	46	5	5
Voice control—signalling traffic, related to call setup, from a voice gateway or a voice application server.	AF31	26	3	3
Video conferencing—in most networks, video conferencing over IP has similar loss, delay, and delay variation requirements as voice over IP traffic.	AF41	34	4	4
Streaming video—relatively high bandwidth applications with a high tolerance for loss, delay, and delay variation. Usually considered more important than regular background applications such as e-mail and web browsing.	AF13	14	1	1
Mission critical date (gold data)—delay-sensitive applications critical to the operation of an enterprise. Level 1 Level 2 Level 3	AF21 AF22 AF23	18 20 22	2 2 2	2 2 2

Table 14-1 Typical Traffic Classifications (continued)

Traffic Type	DSCP per-hop	DSCP (decimal)	IP Precedence	CoS
Less critical data (silver data)—noncritical, but relatively important data.				
Level 1	AF11	10	1	1
Level 2	AF12	12	1	1
Level 3	AF13	14	1	1
Best-effort data (bronze data)—other traffic, including all noninteractive traffic, regardless of importance.	Default	0	0	0
Less than best-effort data—noncritical, bandwidth-intensive data traffic given the least preference. This is the first traffic type to be dropped.				
Level 1		2	0	0
Level 2		4	0	0
Level 3		6	0	0

Classification Based on QoS ACLs

Packets can also be classified in input policy maps based on an ACL lookup. The ACL classification is communicated to an output policy by assigning a QoS group or number in the input policy map. To classify based on ACL lookup, you first create an IP or MAC ACL. Configure a class map and use the **match access-group** {*acl-number* | *acl name*} class-map configuration command, and attach the class map to a policy map.

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

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

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

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

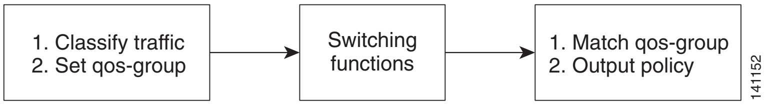
Note Only one access-group is supported per class for an input policy map.

Classification Based on QoS Groups

A QoS group is an internal label used by the switch module 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 module that sets the label. QoS groups provide a way to tag a packet for subsequent QoS action without explicitly marking (changing) the packet. You can then communicate an ACL match from an input policy map to an output policy map.

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 14-4). The QoS groups help aggregate different classes of input traffic for a specific action in an output policy.

Figure 14-4 QoS Groups



You can use QoS groups to aggregate multiple input streams across input classes and policy maps for 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 to the QoS group number in the output policy map to specify the required queuing and scheduling actions.

You can also use QoS groups to identify traffic entering a particular interface if the traffic must be treated differently at the output based on the input interface.

You can use QoS groups to configure per-port, per-VLAN QoS output policies on the egress interface for bridged traffic on the VLAN. Assign a QoS group number to a VLAN on the ingress interface by configuring a per-port, per-VLAN input policy. Then use the same QoS-group number for classification at the egress. Because the VLAN of bridged traffic does not change during forwarding through the switch module, the QoS-group number assigned to the ingress VLAN can be used on the egress interface to identify the same VLAN.

You can use the **cpu traffic qos** [**cos** *value* | **dscp** *value* | **precedence** *value* | **qos-group** *value*] global configuration command to configure a QoS group number for CPU-generated traffic.

Independently you can assign QoS-group numbers at the ingress to any combination of interfaces, VLANs, traffic flows, and aggregated traffic. To assign QoS-group numbers, configure a QoS group marking in an input policy map, along with any other marking or policing actions required in the input policy map for the same service class. This allows the input marking and policing functions to be decoupled from the egress classification function if necessary because only the QoS group must be used for egress classification.

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 subsequently used in an output policy with no mark or change to the packet. You use the **match qos-group** in the output policy.

Note 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 module supports a maximum of 100 QoS groups.

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. If you try to attach an input per-port, per VLAN hierarchical policy to a port that is not a trunk port, the configuration is rejected.

The switch module supports two policy levels: a *parent* level and a *child* level. With the QoS parent-child structure, you can reference a child policy in a parent policy to provide additional control of a specific traffic type. For per-port, per-VLAN QoS, the parent-level class map specifies only the VLAN match criteria, and the child-level class maps provide more detailed classification for frames matching the parent-level class map.You can configure multiple service classes at the parent level to match different combinations of VLANs, and you can apply independent QoS policies to each parent service class using any child policy map.

Note A per-port, per-VLAN parent-level class map supports only a child-policy association; it does not allow any actions to be configured. In addition, for a parent-level class map, you cannot configure an action or a child-policy association for the class **class-default**.

Per-port, per-VLAN QoS has these limitations:

- You can apply a per-port, per-VLAN hierarchical policy map only to trunk ports.
- You can configure classification based on VLAN ID only in the parent level of a per-port, per-VLAN hierarchical policy map.
- When the child policy map attached to a VLAN or set of VLANs contains only Layer 3 classification (**match ip dscp**, **match ip precedence**, **match IP ACL**), you must be careful to ensure that these VLANs are not carried on any port other than the one on which this per-port, per-VLAN policy is attached. Not following this restriction could result in improper QoS behavior for traffic ingressing the switch module on these VLANs.
- We also recommend that you restrict VLAN membership on the trunk ports to which the per-port, per-VLAN is applied by using the **switchport trunk allowed vlan** interface configuration command. Overlapping VLAN membership between trunk ports that have per-port, per-VLAN policies with Layer 3 classification could also result in unexpected QoS behavior.

In this example, the class maps in the child-level policy map specify matching criteria for voice, data, and video traffic, and the child policy map sets the action for input policing each type of traffic. The parent-level policy map specifies the VLANs to which the child policy maps are applied on the specified port:

```
Switch(config)# class-map match-any dscp-1 data
Switch(config-cmap)# match ip dscp 1
Switch(config-cmap)# exit
Switch(config)# class-map match-any dscp-23 video
Switch(config-cmap)# match ip dscp 23
Switch(config-cmap)# exit
Switch(config)# class-map match-any dscp-63 voice
Switch(config-cmap)# match ip dscp-63
Switch(config-cmap)# exit
Switch(config)# class-map match-any customer-1-vlan
Switch(config-cmap)# match vlan 100
Switch(config-cmap)# match vlan 200
Switch(config-cmap)# match vlan 300
Switch(config-cmap)# exit
```

Note You can also enter the match criteria as **match vlan 100 200 300** with the same result.

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

Switch(config)# policy-map parent-customer-1
Switch(config-pmap)# class customer-1-vlan
Switch(config-pmap-c)# service-policy ingress-policy-1
Switch(config-pmap-c)# exit

Switch(config)# interface fastethernet0/1
Switch(config-if)# switchport mode trunk
Switch(config-if)# service-policy input customer-1-ingress
Switch(config-pmap-c)# exit
```

Note Each per-port, per-VLAN parent policy class, except **class-default**, can have a child policy association.

See the "Configuring Per-Port Per-VLAN QoS with Hierarchical Input Policy Maps" section on page 14-59 for configuration information, including configuration guidelines and limitations.

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 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 module 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 module:

- 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 **set** command in a policy map or police function. Individual policers also support the **violate-action** command, but aggregate policers do not support table maps with **violate-action**.

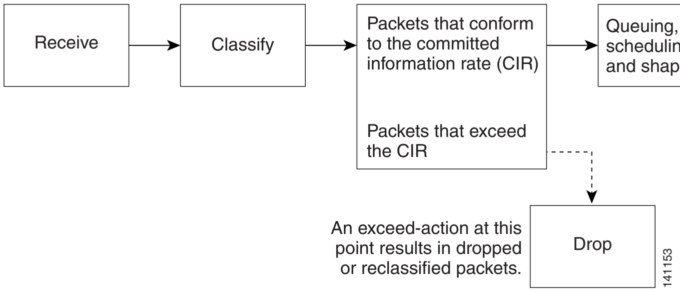
Table maps are not supported in output policy maps.

Policing

After a packet is classified, you can use policing as shown in Figure 14-5 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 classes. See the "Unconditional Priority Policing" section on page 14-20.

Figure 14-5 Policing of Classified Packets


Receive
Classify
Packets that conform to the committed information rate (CIR)
Packets that exceed the CIR
Queuing, scheduling, and shaping
An exceed-action at this point results in dropped or reclassified packets.
Drop
141153

These sections describe the types of policing supported on the switch module:

- Individual Policing, page 14-16
- Aggregate Policing, page 14-18
- Unconditional Priority Policing, page 14-20

Individual Policing

Individual policing applies only to input policy maps. In policy-map configuration mode, you enter the **class** command followed by class-map name, and enter policy-map class configuration mode.

The switch module supports 1-rate, 2-color ingress policing and 2-rate, 3-color policing for individual or aggregate policing.

For 1-rate, 2-color policing, you 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. For more information, see the "Attaching a Traffic Policy to an Interface" section on page 14-47.

When you configure a 2-rate policer, in addition to configuring the committed information rate (CIR) for updating the first token bucket, you also configure 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 optionally set actions to perform on packets that conform to the specified CIR and PIR (conform-action), packets that conform to the PIR, but not the CIR (exceed-action), and packets that exceed the PIR value (violate-action).

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

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

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

This is an example of basic policing for all traffic received with a CoS of 4. The first value following the **police** command limits the average traffic rate to 10, 000,000 bits per second (bps); the second value represents the additional burst size (10 kilobytes). The policy is assigned to Fast Ethernet port 1:

```
Switch(config)# table-map cos-dscp-tablemap
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-police)# exit
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface fastethernet0/1
Switch(config-if)# service-policy input video-policy
Switch(config-if)# exit
```

You can use the **conform-action**, **exceed-action**, and **violate-action** policy-map class configuration commands or the **conform-action**, **exceed-action**, and **violate-action** policy-map class police configuration commands to specify the action to be taken 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 configure each marking action by using explicit values, table maps, or a combination of both. Table maps list specific traffic attributes and map (or convert) them to other attributes.

You can configure multiple conform, exceed, and violate actions simultaneously for each service class. If you do not configure a **violate-action**, by default the violate class is assigned the same action as the **exceed-action**.

After you create a table map, you configure a policy-map policer to use the table map.

Note When you use a table map in an input policy map, the protocol type for the **from**–action in the table map must be the same as the protocol type of the associated classification. For example, if a class map represents IP classification, the **from**–type action in the table map must be either **dscp** or **precedence**. If the class map represents a non-IP classification, the **from**–type action in the table map must be **cos**.

To configure multiple actions in a class, you can enter multiple conform, exceed, or violate action entries in policy-map class police configuration mode, as in this example:

```
Switch(config)# policy-map map1
Switch(config-pmap)# class class1
Switch(config-pmap-c)# police 100000 500000
Switch(config-pmap-c-police)# conform-action set-cos-transmit 4
Switch(config-pmap-c-police)# conform-action set-dscp-transmit dscp table
conform-dscp-to-dscp-mutation
Switch(config-pmap-c-police)# conform-action set-qos-transmit 10
Switch(config-pmap-c-police)# exceed-action set-cos-transmit 2
Switch(config-pmap-c-police)# exceed-action set-dscp-transmit dscp table
exceed-dscp-to-dscp-mutation
Switch(config-pmap-c-police)# exceed-action set-qos-transmit 20
Switch(config-pmap-c-police)# exit
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
```

Aggregate Policing

Aggregate policing applies only to input policy maps. An aggregate policer differs from an individual policer because it is shared by multiple traffic classes within a policy map. The switch module supports 1-rate, 2-color ingress policing and 2-rate, 3-color policing for aggregate policing.

You can use the **policer aggregate** global configuration command to set a policer for all traffic received or sent on a physical interface. When you configure an aggregate policer, you can configure specific burst sizes and conform and exceed actions. 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, in addition to configuring the committed information rate (CIR) for updating the first token bucket, you also configure 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 optionally set actions to perform on packets that conform to the specified CIR and PIR (conform-action), packets that conform to the PIR, but not the CIR (exceed-action), and packets that exceed the PIR value (violate-action).

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.

- 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 PIR, the policer is configured as a 1-rate, 2-color policer.

Setting the burst sizes too low can result in less traffic than expected. Setting burst sizes too high can result in more traffic than expected.

You can configure multiple conform and exceed actions simultaneously for each service class. Conform actions are to send the packet without modifications, to set a QoS group value for classification at the egress, or to set a new CoS, DSCP, or IP precedence value. Exceed actions are to drop the packet, to send the packet without modification, to set a QoS group for classification at the egress, or to set a new CoS, DSCP, or IP precedence to a value. You can configure each marking conform or exceed action by using explicit values, using table maps, or using a combination of both. Table maps list specific traffic attributes and map (or convert) them to other attributes.

You can configure multiple conform, exceed, and violate actions simultaneously for each service class. You can use the **conform-action**, **exceed-action**, and **violate-action** policy-map class configuration commands or the **conform-action**, **exceed-action**, and **violate-action** policy-map class police configuration commands to specify the action to be taken 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.

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.

You can configure each marking conform, exceed, or violate action by using explicit values, using table maps, or using a combination of both. If you do not configure a **violate-action**, by default the violate class is assigned the same action as the **exceed-action**.

Table maps list specific traffic attributes and map (or convert) them to other attributes. Table maps are not supported for violate-action for aggregate policing unless you configure a table map for exceed-action and no explicit action is configured for violate action.

After you create a table map, you configure a policy-map policer to use the table map.

Note When you use a table map in an input policy map, the protocol type for the **from**–action in the table map must be the same as the protocol type of the associated classification. For example, if a class map represents IP classification, the **from**–type action in the table map must be either **dscp** or **precedence**. If the class map represents a non-IP classification, the **from**–type action in the table map must be **cos**.

You can configure multiple conform, exceed, and violate actions simultaneously for an aggregate policer as parameters in the **policer aggregate** global configuration command, but you must enter the actions in a particular order. See the configuration guideline in the "Configuring Input Policy Maps with Aggregate Policing" section on page 14-54.

After you configure the aggregate policer, you create a policy map and an associated class map, associate the policy map with the aggregate policer, and apply the service policy to a port.

Note Only one policy map can use any specific aggregate policer. Aggregate policing cannot be used to aggregate traffic streams across multiple interfaces. It can be used only to aggregate traffic streams across multiple classes in a policy map attached to an interface and aggregate streams across VLANs on a port in a per-port, per-VLAN policy map.

After you configure the policy map and policing actions, attach the policy to an ingress port by using the **service-policy** interface configuration command.

The class maps in this example refer to access lists.

```
Switch(config)# policer aggregate agg1 cir 23000 bc 10000 conform-action set-dscp-transmit
46 exceed-action drop
Switch(config)# class-map testclass
Switch(config-cmap)# match access-group 1
Switch(config-cmap)# exit
Switch(config)# class-map videoclass
Switch(config-cmap)# match access-group 2
Switch(config-cmap)# exit
Switch(config)# policy-map testexample
Switch(config-pmap)# class testclass
Switch(config-pmap-c)# police aggregate agg1
Switch(config-pmap-c)# exit
Switch(config-pmap)# class video-class
Switch(config-pmap-c)# police aggregate agg1
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface fastethernet0/1
Switch(config-if)# service-policy input testexample
Switch(config-if)# exit
```

For configuration information, see the "Configuring Input Policy Maps with Aggregate Policing" section on page 14-54.

If the switch module is running the metro IP access or metro access image, you can also use aggregate policing to regulate traffic streams across VLANs, as in this example:

You can also use aggregate policing to regulate traffic streams across VLANs, as in this example:

```
Switch(config)# policer aggregate agg1 cir 23000 bc 10000 conform-action set-dscp-transmit
af31 set-cos-transmit 3 exceed-action set-dscp-transmit af11 set-cos-transmit 1
Switch(config)# class-map video-provider-1
Switch(config-cmap)# match access-group 1
Switch(config-cmap)# exit
Switch(config)# class-map video-provider-2
Switch(config-cmap)# match access-group 2
Switch(config-cmap)# exit
Switch(config)# class-map match-any customer1-provider-100
Switch(config-cmap)# match vlan 100
Switch(config-cmap)# exit
Switch(config)# class-map match-any customer1-provider-200
Switch(config-cmap)# match vlan 200
Switch(config-cmap)# exit
Switch(config)# policy-map child-policy-1
Switch(config-pmap)# class video-provider-1
Switch(config-pmap-c)# set dscp af41
Switch(config-pmap-c)# exit
Switch(config-pmap)# class class-default
Switch(config-pmap-c)# police aggregate agg1
Switch(config-pmap-c)# exit
Switch(config)# policy-map child-policy-2
Switch(config-pmap)# class video-provider-2
Switch(config-pmap-c)# set dscp cs4
Switch(config-pmap-c)# exit
Switch(config-pmap)# class class-default
Switch(config-pmap-c)# police aggregate agg1
Switch(config-pmap-c)# exit
Switch(config)# policy-map customer-1-ingress
Switch(config-pmap)# class customer1-provider-100
Switch(config-pmap-c)# service-policy child-policy-1
Switch(config-pmap-c)# exit
Switch(config-pmap)# class customer1-provider-200
Switch(config-pmap-c)# service-policy child-policy-2
Switch(config-pmap-c)# exit
Switch(config)# interface fastethernet0/1
Switch(config-if)# switchport mode trunk
Switch(config-if)# service-policy input customer-1-ingress
Switch(config-pmap-c)# exit
```

Unconditional 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, at the expense of other queues. Excessive use of high-priority queuing can create congestion for lower priority traffic.

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.

Note You can configure 1-rate, 2-color policers for output policy maps with priority. You cannot configure 2-rate, 3-color policers for output policies.

See also the "Configuring Output Policy Maps with Class-Based Priority Queuing" section on page 14-72.

Note You cannot configure a policer committed burst size for an unconditional priority policer. Any configured burst size is ignored.

This example shows how to use the **priority** with **police** commands to configure *out-class1* as the priority queue, with traffic going to the queue 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 **class-default** queue gets the remaining port bandwidth.0

```
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
```

Marking

You can use packet marking in input policy maps to set or modify the attributes for traffic belonging to a specific class. After network traffic is organized into classes, you use marking to identify certain traffic types for unique handling. 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 module.

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.

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

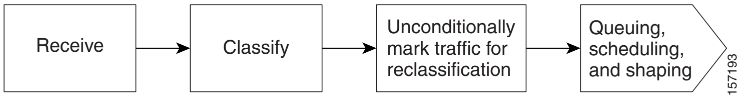
You can also mark traffic by using the **set** command with table maps.Table maps list specific traffic attributes and maps (or converts) them to another attribute. A table map establishes a to-from relationship for the attribute and defines the change to be made.

You can simultaneously configure actions to modify DSCP, precedence, and COS markings in the packet for the same service along with QoS group marking actions. You can use the QoS group number defined in the marking action for egress classification.

Note When you use a table map in an input policy map, the protocol type of the **from**-type action in the table map must be the same as the protocol type of the associated classification. For example, if a class map represents an IP classification, the **from**-type action in the table map must be either **dscp** or **precedence**. If the class map represents a non-IP classification, the **from**-type action in the table map must be **cos**.

After you create a table map, you configure a policy map to use the table map. See the "Congestion Management and Scheduling" section on page 14-26. Figure 14-6 shows the steps for marking traffic.

Figure 14-6 Marking of Classified Traffic


Receive
Classify
Unconditionally mark traffic for reclassification
Queuing, scheduling, and shaping
157193

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 fastethernet0/1
Switch(config-if)# service-policy input Example
Switch(config-if)# exit
```

QoS Treatment for Performance-Monitoring Protocols

- Configuration Guidelines, page 14-25

Cisco IP-SLAs

For information about Cisco IP service level agreements (IP-SLAs), see "Understanding Cisco IOS IP SLAs."

QoS Treatment for IP-SLA and TWAMP Probes

The QoS treatment for IP-SLA and TWAMP probes must exactly reflect the effects that occur to the normal data traffic crossing the device.

The generating device should not change the probe markings. It should queue these probes based on the configured queueing policies for normal traffic.

Marking

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

By default, IP traffic marking (including IP SLA and TWAMP probes) is not changed. This feature can change this behavior.

Queuing

The CFM traffic (including IP SLAs using CFM probes) is queued according to its CoS value and the output policy map configured on the egress port, similar to normal traffic. This feature cannot change this behavior.

IP traffic (including IP SLA and TWAMP probes) is queued according to the markings specified in the **cpu traffic qos** global configuration command and the output policy map on the egress port. If this command is not configured, all IP traffic is statically mapped to a queue on the egress port.

QoS Marking for CPU-Generated Traffic

You can use QoS marking to set or modify the attributes of traffic from the CPU. The QoS marking action can cause the CoS, DSCP, or IP precedence bits in the packet to be rewritten or left unchanged. QoS uses packet markings to identify certain traffic types and how to treat them on the local switch module and the network.

You can also use marking to assign traffic to a QoS group within the switch module. This QoS group is an internal label that does not modify the packet, but it can be used to identify the traffic type when configuring egress queuing on the network port.

You can specify and mark traffic CPU-generated traffic by using these global configuration commands:

cpu traffic qos cos {*cos_value* | **cos** [**table-map** *table-map-name*] | **dscp** [**table-ma**p *table-map-name*] | **precedence** [**table-map** *table-map-name*]}

cpu traffic qos dscp {*dscp_value* | **cos** [**table-map** *table-map-name*] | **dscp** [**table-map** *table-map-name*] | **precedence** [**table-map** *table-map-name*]}

cpu traffic qos precedence {*precedence_value* | **cos** [**table-map** *table-map-name*] | **dscp** [**table-map** *table-map-name*] | **precedence** [**table-map** *table-map-name*]}

cpu traffic qos qos-group *value*

You can mark CoS, IP-DSCP, IP precedence, and QoS group by configuring an explicit value or by using the **table-map** keyword. Table maps list specific traffic attributes and map (or convert) them to another attribute. A table map establishes a to-from relationship for the attribute and defines the change to be made:

- Marking CoS by using the CoS, or the IP-DSCP, or the IP precedence of IP CPU-packets
- Marking CoS by using the CoS of non-IP CPU-packets.
- Marking IP DSCP by using the CoS, or the IP-DSCP, or the IP precedence of the CPU-packet
- Marking IP precedence by using the CoS, or the IP-DSCP, or the IP precedence of the CPU-packet

You can configure either IP-DSCP or IP precedence marking.

You can also simultaneously configure marking actions to modify CoS, IP-DSCP or IP precedence, and QoS group.

The **cpu traffic qos** command specifies the traffic to which it applies: all CPU traffic, only CPU IP traffic, or only CPU non-IP traffic. All other traffic retains its QoS markings. This feature does not affect CFM traffic (including Layer 2 IP SLA probes using CFM).

Note The switch module provides the ability to mark CoS, IP-DSCP and IP precedence of CPU-generated traffic by using table maps.

QoS Queuing for CPU-Generated Traffic

You can use the QoS markings established for the CPU-generated traffic by the **cpu traffic qos** global configuration command as packet identifiers in the class-map of an output policy-map to map CPU traffic to class-queues in the output policy-map on the egress port. You can then use output policy-maps on the egress port to configure queuing and scheduling for traffic leaving the switch module from that port.

If you want to map *all* CPU-generated traffic to a single class in the output policy-maps without changing the CoS, IP DSCP, or IP-precedence packet markings, you can use QoS groups for marking CPU-generated traffic.

If you want to map *all* CPU-generated IP traffic to classes in the output policy maps based on IP-DSCP or IP precedence without changing those packet markings, you can use a table map:

- Configure IP-DSCP or IP precedence marking by using **DSCP** or **precedence** as the **map from** value *without* a table map.
- Configure IP-DSCP or IP-precedence marking by using **DSCP** or **precedence** as the **map from** value *with* a table map, using only the **default** and **copy** keywords.

If you want to map *all* CPU-generated traffic to classes in the output policy maps based on the CoS without changing the CoS packet markings, you can use the table map:

- Configure CoS marking by using **CoS** as the **map from** value *without* a table map.
- Configure CoS marking using **CoS** as the **map from** value *with* a table map, using only the **default** and **copy** keywords.

For details about table maps, see the "Table Maps" section on page 14-14.

Using the **cpu traffic qos** global configuration command with table mapping, you can configure multiple marking and queuing policies to work together or independently. You can queue native VLAN traffic based on the CoS markings configured using the **cpu traffic qos** global configuration command.

The **cpu traffic qos** command specifies the traffic to which it applies: all CPU traffic, only CPU-IP traffic, or only CPU non-IP traffic. All other traffic is statically mapped to a CPU-default queue on the egress port. All CFM traffic (including Layer 2 IP SLA probes using CFM) is mapped to classes in the output policy map and queued based on their CoS value.

Note The switch module provides the ability to queue based on the CoS, IP-DSCP, and IP precedence of CPU-generated traffic.

Configuration Guidelines

- This feature must be configured globally for a switch module; it cannot be configured per-port or per-protocol.
- Enter each **cpu traffic qos** marking action on a separate line.
- The **cpu traffic qos cos** global configuration command configures CoS marking for CPU-generated traffic by using either a specific CoS value or a table map, but not both. A new configuration overwrites the existing configuration.
- The **cpu traffic qos dscp** global configuration command configures IP-DSCP marking for CPU-generated IP traffic by using either a specific DSCP value or a table map, but not both. A new configuration overwrites the existing configuration.
- The **cpu traffic qos precedence** global configuration command configures IP-precedence marking for CPU-generated IP traffic by using either a specific precedence value or a table map, but not both. A new configuration overwrites the existing configuration.
- The **cpu traffic qos dscp** and **cpu traffic qos precedence** global configuration commands are mutually exclusive. A new configuration overwrites the existing configuration.
- When the **cpu traffic qos dscp** global configuration command is configured with table maps, you can configure only one **map from** value at a time—DSCP, precedence, or CoS. A new configuration overwrites the existing configuration. Packets marked by this command can be classified and queued by an output policy map based on the marked DSCP or precedence value.
- When the **cpu traffic qos precedence** global configuration command is configured with table maps, you can configure only one **map from** value at a time—DSCP, precedence, or CoS. A new configuration overwrites the existing configuration. Packets marked by this command can be classified and queued by an output policy map based on the marked precedence or DSCP value.
- You cannot configure a **map from** value of both DSCP and precedence. A new configuration overwrites the existing configuration.
- When the **cpu traffic qos cos** global configuration command is configured with table maps, you can configure two **map from** values at a time—CoS and either DSCP or precedence.
- If the **cpu traffic qos cos** global configuration command is configured with only a **map from** value of DSCP or precedence:
  - The CoS value of IP packets is mapped by using the DSCP (or precedence) value in the packet and the configured table map. Packets can be classified and queued by an output policy map based on the marked CoS value.
  - The CoS value of non-IP packets remains unchanged.

- If the **cpu traffic qos cos** global configuration command is configured with a **map from** value of CoS:
  - The CoS value of IP packets is mapped by using the CoS value in the packet and the configured table map. Packets can be classified and queued by an output policy map based on the marked CoS value.
  - The CoS value of non-IP packets is mapped by using the CoS value in the packet and the configured table map. Packets can be classified and queued by an output policy map based on the marked CoS value.
- If the **cpu traffic qos cos** global configuration command is configured with a **map from** value of DSCP or precedence and CoS:
  - The CoS value of IP packets is mapped by using the DSCP or precedence value in the packet and the configured table map. Packets can be classified and queued by an output policy map based on the marked CoS value.
  - The CoS value of non-IP packets is mapped by using the CoS value in the packet and the configured table map. Packets can be classified and queued by an output policy map based on the marked CoS value.
- The **cpu traffic qos qos-group** global configuration command can be used to configure QoS group marking for CPU-generated traffic only for a specific QoS group. The **table-map** option is not available.

Congestion Management and Scheduling

Cisco Modular QoS CLI (MQC) provides several related mechanisms to control outgoing traffic flow. They are implemented in output policy maps to control output traffic queues. The scheduling stage holds packets until the appropriate time to send them to one of the four 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 module 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 a percentage of total bandwidth or of remaining bandwidth.
- Priority queuing or class-based priority queuing

  You use the **priority** policy-map class configuration command to specify the priority of a type of traffic over other types of traffic. You can specify strict priority for the high-priority traffic and allocate any excess bandwidth to other traffic queues, or specify priority with unconditional policing of high-priority traffic and allocate the known remaining bandwidth among the other traffic queues.

- To configure strict priority, use only the **priority** policy-map class configuration command to configure the priority queue. Use the **bandwidth remaining percent** policy-map class configuration command for the other traffic classes to allocate the excess bandwidth in the desired ratios.
- To configure priority with unconditional policing, configure the priority queue by using the **priority** policy-map class configuration command and the **police** policy-map class configuration command to unconditionally rate-limit the priority queue. In this case, you can configure the other traffic classes with **bandwidth** or **shape average**, depending on requirements.

These sections contain additional information about scheduling:

- Traffic Shaping, page 14-27
- Class-Based Weighted Fair Queuing, page 14-29
- Priority Queuing, page 14-30

Traffic Shaping

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

Note You cannot configure traffic shaping (**shape average**) and CBWFQ (**bandwidth**) or priority queuing (**priority**) for the same class in an output policy map. You cannot configure traffic shaping for a traffic class when strict priority (priority without police) is configured for another class within the output policy-map.

Class-Based Shaping

Class-based shaping uses the **shape average** policy-map class configuration command to limit the rate of data transmission as the number of bits per second to be used for the committed information rate for a class of traffic. The switch module supports separate queues for three classes of traffic. The fourth queue is always the default queue for class **class-default**, unclassified traffic.

Note On the switch module, configuring traffic shaping also automatically sets the minimum bandwidth guarantee or committed information rate (CIR) of the queue to the same value as the PIR.

This example shows how to configure traffic shaping for outgoing traffic on a Fast Ethernet port so that *outclass1*, *outclass2*, and *outclass3* get a maximum of 50, 20, and 10 Mbps, respectively, of the available port bandwidth. The class **class-default** at a minimum gets the remaining 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 fastethernet 0/1
Switch(config-if)# service-policy output out-policy
Switch(config-if)# exit
```

Port Shaping

To configure port shaping (a transmit port shaper), create a policy map that contains only a default class, and use the **shape average** command to specify the maximum bandwidth for a port.

This example shows how to configure a policy map that shapes a port to 90 Mbps, allocated according to the *out-policy* policy map configured in the previous example. The **service-policy** policy map class command is used to create a child policy to the parent:

```
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 fastethernet0/1
Switch(config-if)# service-policy output out-policy-parent
Switch(config-if)# exit
```

Parent-Child Hierarchy

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

The first policy level, the parent level, is used for port shaping, and you can specific only one class of type **class-default** within the policy. This is an example of a parent-level policy map:

```
Switch(config)# policy-map parent
Switch(config-pmap)# class class-default
Switch(config-pmap-c)# shape average 50000000
Switch(config-pmap-c)# exit
```

The second policy level, the *child* level, is used to control a specific traffic stream or class, as in this example:

```
Switch(config)# policy-map child
Switch(config-pmap)# class class1
Switch(config-pmap-c)# priority
Switch(config-pmap-c)# exit
```

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 fastethernet0/1
Switch(config-if)# service-policy output parent
Switch(config-if)# 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 output bandwidth 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.

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

  Note You cannot configure bandwidth as an absolute rate or a percentage of total bandwidth when strict priority (priority without police) is configured for another class in the output policy.

- When you use the **bandwidth** policy-map class configuration command to configure a class of traffic as a percentage of *remaining* bandwidth, this represents the portion of the excess bandwidth of the port that is allocated to the class. This means that the class is allocated bandwidth only if there is excess bandwidth on the port, and if there is no minimum bandwidth guarantee for this traffic class.

  Note You can configure bandwidth as percentage of remaining bandwidth only when strict priority (priority without police) is configured for another class in the output policy map.

For more information, see the "Configuring Output Policy Maps with Class-Based-Weighted-Queuing" section on page 14-67.

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

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 kbps. Any excess bandwidth is divided among the classes in the same proportion as the CIR rated:

```
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 fastethernet 0/1
Switch(config-if)# service-policy output out-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 also not eligible for any excess bandwidth and as a result receives no bandwidth.

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 fastethernet 0/1
Switch(config-if)# service-policy output out-policy
Switch(config-if)# exit
```

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 strict priority queuing, the priority queue is constantly serviced. All packets in the queue are scheduled and sent until the queue is empty. Priority queuing allows traffic for the associated class to be sent before packets in other queues are sent.

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

The switch module supports strict priority queuing or priority used with the **police** policy-map command.

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

  Note You cannot configure priority without policing for a traffic class when traffic shaping or CBWFQ are configured for another class in the same output policy map.

- You can use priority with the **police** policy-map command, or *unconditional priority policing*, to reduce the bandwidth used by the priority queue. This is the only form of policing that is supported in output policy maps. Using this combination of commands configures a maximum rate on the priority queue, and you can use the **bandwidth** and **shape average** policy-map commands for other classes to allocate traffic rates on other queues.

  Note 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.

Priority queuing has these restrictions:

- You can associate the **priority** command with a single unique class for all attached output polices on the switch module.
- 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 Output Policy Maps with Class-Based Priority Queuing" section on page 14-72.

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 gigabitethernet 0/1
Switch(config-if)# service-policy output policy1
Switch(config-if)# exit
```

This example shows how to use the **priority** with **police** commands to configure *out-class1* as the priority queue, with traffic going to the queue limited to 20000000 bps so that the priority queue will never use more than that. Traffic above that rate is dropped. The other traffic queues are configured to use 50 and 20 percent of the bandwidth that is left, as in the previous example.

```
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 percent 50
Switch(config-pmap-c)# exit
Switch(config-pmap)# class out-class3
Switch(config-pmap-c)# bandwidth percent 20
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit

Switch(config)# interface gigabitethernet 0/1
Switch(config-if)# service-policy output policy1
Switch(config-if)# exit
```

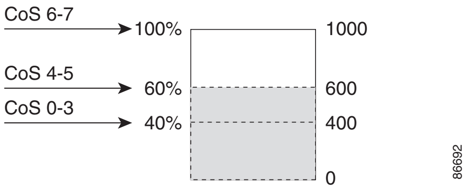
Congestion Avoidance and Queuing

Congestion avoidance 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 module uses Weighted Tail Drop (WTD) to manage the queue sizes and provide a drop precedence for traffic classifications. You set the queue size limits depending on the markings of the packets in the queue. Each packet that travels through the switch module can be assigned to a specific queue and threshold. For example, specific DSCP or CoS values can be mapped to a specific egress queue and threshold.

WTD is implemented on traffic queues to manage the queue size and to provide drop precedences for different traffic classifications. As a frame enters a particular queue, WTD uses the packet classification to subject it to different thresholds. If the total destination queue size is greater than the threshold of any reclassified traffic, the next frame of that traffic is dropped.

Figure 14-7 shows an example of WTD operating on a queue of 1000 frames. Three drop percentages are configured: 40 percent (400 frames), 60 percent (600 frames), and 100 percent (1000 frames). These percentages mean that traffic reclassified to the 40-percent threshold is dropped when the queue depth exceeds 400 frames, traffic reclassified to 60 percent is dropped when the queue depth exceeds 600 frames, and traffic up to 400 frames can be queued at the 40-percent threshold, up to 600 frames at the 60-percent threshold, and up to 1000 frames at the 100-percent threshold.

Figure 14-7 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 frames, and a new frame arrives containing CoS values 4 and 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 module drops it.

WTD is configured by using the **queue-limit** policy-map class command. The command adjusts the queue size (buffer size) associated with a particular class of traffic. You specify the threshold as the number of packets, where each packet is a fixed unit of 256 bytes. You can specify different queue sizes for different classes of traffic (CoS, DSCP, precedence, 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 module supports up to three unique queue-limit configurations across all output policy maps. Within an output policy map, only four queues (classes) are allowed, including the class default. Each queue has three thresholds defined. Only three unique threshold value configurations are allowed on the switch module. However, multiple policy maps can share the same queue-limits. When two policy maps a share queue-limit configuration, all threshold values must be the same for all the classes in both policy maps.

For more information, see the "Configuring Output Policy Maps with Class-Based-Weighted-Queuing" section on page 14-67.

This example configures *class A* to match DSCP values and a policy map, *PM1*. The DSCP values of 30 and 50 are mapped to unique thresholds (32 and 64, respectively). The DSCP values of 40 and 60 are mapped to the maximum threshold of 112 packets.

```
Switch(config)# class-map match-any classA
Switch(config-cmap)# match ip dscp 30 40 50 60
Switch(config-cmap)# exit
Switch(config)# policy-map PM1
Switch(config-pmap)# class classA
Switch(config-pmap-c)# bandwidth percent 50
Switch(config-pmap-c)# queue-limit 112
Switch(config-pmap-c)# queue-limit dscp 30 32
Switch(config-pmap-c)# queue-limit dscp 50 64
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet 0/1
Switch(config-if)# service-policy output PM1
Switch(config-if)# exit
```

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

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

Note When you configure a queue limit for a class in an output policy map, all other output policy maps must use the same qualifier type and qualifier value format. Only the queue-limit threshold values can be different. For example, when you configure *class A* queue limit thresholds for **dscp 30** and **dscp 50** in policy map *PM1*, and you configure *class A* queue limits in policy map *PM2*, you must use **dscp 30** and **dscp 50** as qualifiers. You cannot use **dscp 20** and **dscp 40**. The threshold values can be different, but different threshold values creates a new queue-limit configuration.

By default, the total amount of buffer space is divided equally among all ports and all queues per port, which is adequate for many applications. You can decrease the queue size for latency-sensitive traffic or increase the queue size for bursty traffic.

Note When you use the **queue-limit** command to configure queue thresholds for a class, the WTD thresholds must be less than or equal to the queue maximum threshold. A queue size configured with no qualifier must be larger than any queue sizes configured with qualifiers.

When you configure queue limit, the range for the number of packets is from 16 to 544, in multiples of 16, where each packet is a fixed unit of 256 bytes.

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

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.

Note These restrictions apply to WTD qualifiers:

- You cannot configure more than two threshold values for WTD qualifiers (**cos**, **dscp**, **precedence**, **qos-group**) by using the **queue-limit** command. However, there is no limit to the number of qualifiers that you can map to these thresholds. You can configure a third threshold value to set the maximum queue by using the **queue-limit** command with no qualifiers.
- A WTD qualifier in the **queue-limit** command must be the same as at least one **match** qualifier in the associated class map.

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

```
Switch(config)# policy-map out-policy
Switch(config-pmap)# class outclass1
Switch(config-pmap-c)# bandwidth percent 40
Switch(config-pmap-c)# queue-limit 48
Switch(config-pmap-c)# exit
Switch(config-pmap)# class outclass2
Switch(config-pmap-c)# bandwidth percent 20
Switch(config-pmap-c)# queue-limit 32
Switch(config-pmap-c)# exit
Switch(config-pmap)# class outclass3
Switch(config-pmap-c)# bandwidth percent 10
Switch(config-pmap-c)# queue-limit 16
Switch(config-pmap-c)# exit
Switch(config-pmap)# class class-default
Switch(config-pmap-c)# bandwidth percent 10
Switch(config-pmap-c)# queue-limit 272
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet 0/1
Switch(config-if)# service-policy output out-policy
Switch(config-if)# exit
```

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

Configuring QoS

Before configuring QoS, you must have a thorough understanding of these factors:

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

These sections describe how to classify, police, and mark incoming traffic, and schedule and queue outgoing traffic. Depending on your network configuration, you must perform one or more of these tasks.

- Default QoS Configuration, page 14-35
- QoS Configuration Guidelines, page 14-36
- Using ACLs to Classify Traffic, page 14-37
- Using Class Maps to Define a Traffic Class, page 14-42
- Configuring Table Maps, page 14-45
- Attaching a Traffic Policy to an Interface, page 14-47
- Configuring Input Policy Maps, page 14-47
- Configuring Output Policy Maps, page 14-66
- Configuring QoS Marking and Queuing for CPU-Generated Traffic, page 14-80

Default QoS Configuration

There are no policy maps, class maps, table maps, or policers configured. At the egress port, all traffic goes through a single default queue that is given the full operational port bandwidth. The default size of the default queue is 160 (256-byte) packets.

The packets are not modified (the CoS, DSCP, and IP precedence 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.

QoS Configuration Guidelines

- You can configure QoS only on physical ports.
- On a port configured for QoS, all traffic received through the port is classified, policed, and marked according to the input policy map attached to the port. On a trunk port configured for QoS, traffic in all VLANs received through the port is classified, policed, and marked according to the policy map attached to the port. If a per-port, per-VLAN policy map is attached, traffic on the trunk port is classified, policed, and marked for the VLANs specified in the parent-level policy, according to the child policy map associated with each VLAN.
- If you have EtherChannel ports configured on your switch module, you must configure QoS classification, policing, mapping, and queuing on the individual physical ports that comprise the EtherChannel. You must decide whether the QoS configuration should match on all ports in the EtherChannel.
- Control traffic (such as spanning-tree bridge protocol data units [BPDUs] and routing update packets) received by the switch module are subject to all ingress QoS processing.
- You are likely to lose data when you change queue settings; try to make changes when traffic is at a minimum.
- When you try to attach a new policy to an interface and this brings the number of policer *instances* to more than 1,024 minus 1 more than the number of interfaces on the switch module, you receive an error message, and the configuration fails.
- When you try to attach new policy to an interface, increasing the number of policer profiles to more than 256, you receive an error message, and the configuration fails. A profile is a combination of commit rate, peak rate, commit burst, and peak burst. You can attach one profile to multiple instances, but if one of these characteristics differs, the policer is considered to have a new profile.
- You can specify 256 *unique* VLAN classification criteria within a per-port, per-VLAN policy-map, across all ports on the switch module. Any policy attachment or change that causes this limit to be exceeded fails with a *VLAN label resources exceeded* error message.
- You can attach per-port and per-port, per-VLAN policy-maps across all ports on the switch module until QoS ACE classification resource limitations are reached. Any policy attachment or change that causes this limit to be exceeded fails with a *TCAM resources exceeded* error message.
- When CPU protection is enabled, you can configure only 45 policers per port. Disabling CPU protection allows you to configure up to 64 policers per port. You can enter the **show policer cpu uni-eni** {**drop** | **rate**} privileged EXEC command to see if CPU protection is enabled.
- Note these limitations when you disable CPU protection:
  - When CPU protection is disabled, you can configure a maximum of 63 policers per port for user-defined classes, and one for class-default. Any policy attachment or change that causes this limit to be exceeded fails with a *policer resources exceeded* error message.
  - When CPU protection is disabled, you can configure a maximum of 256 policers on the switch modulefor ME3400-24TS ME3400E-24TS platform. Any policy attachment or change that causes this limit to be exceeded fails with a *policer resources exceeded* error message.
  - If you disable CPU protection and attach a policy map with more than 45 policers, and then enable CPU protection again, and reload, 19 policers per port are again required for CPU protection. During reload, the policers 46 and above will reach the *policer resources exceeded* error condition and no policers are attached to those classes.
- If the number of internal QoS labels exceeds 256, you receive an error message.

- Table maps are not supported for **violate-action** for aggregate policing unless you configure a table map for exceed-action and no explicit action is configured for **violate-action**. For both individual and aggregate policers, if you do not configure a **violate-action**, by default the violate class is assigned the same action as the **exceed-action**.
- If double-tagged packets are received on a trunk or 802.1Q tunnel interface, these packets can be classified on DSCP and IP precedence along with other parameters, but you cannot set DSCP or IP precedence on the outgoing packets. You can set CoS on the outgoing packets.

See the configuration sections for specific QoS features for more configuration guidelines related to each feature.

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 Chapter 34, “Configuring Network Security with ACLs" in the *Cisco CGS 2520 Switch Software Configuration Guide*:

http://www.cisco.com/en/US/docs/switches/connectedgrid/cgs2520/software/release/12_2_53_ex/configuration/guide/swacl.html

Follow these guidelines when configuring QoS ACLs:

- You cannot match IP fragments against configured IP extended ACLs to enforce QoS. IP fragments are sent as best-effort. IP fragments are denoted by fields in the IP header.
- The switch module supports only one access group per class in an input policy map.
- You cannot configure **match-access** group in an output policy map.

These sections describe how to create QoS ACLs:

- “Creating IP Standard ACLs” section on page 14-38
- “Creating IP Extended ACLs” section on page 14-39
- “Creating Layer 2 MAC ACLs” section on page 14-40

Creating IP Standard ACLs

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

	Step	Command
Step 1	Enter global configuration mode.	**configure terminal**
Step 2	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.	**access-list** *access-list-number* **permit** *source* [*source-wildcard*]
Step 3	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*]	**ip access-list standard** *name*
Step 4	Return to privileged EXEC mode.	**end**
Step 5	Verify your entries.	**show access-lists**

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

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

	Step	Command
Step 1	Enter global configuration mode.	**configure terminal**
Step 2	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. Use the question mark (?) to see a list of available protocols. To match any Internet protocol (including ICMP, TCP, and UDP), enter **ip**. • The *source* is the number of the network or host sending the packet. • The *source-wildcard* applies wildcard bits to the source. • The *destination* is the network or host number receiving the packet. • The *destination-wildcard* applies wildcard bits to the destination. You can specify source, destination, and wildcards as: • The 32-bit quantity in dotted-decimal format. • The keyword **any** for 0.0.0.0 255.255.255.255 (any host). • The keyword **host** for a single host 0.0.0.0. Other keywords are optional and have these meanings: • **precedence**—Enter to match packets with a precedence level specified as a number from 0 to 7 or by name: **routine** (**0**), **priority** (**1**), **immediate** (**2**), **flash** (**3**), **flash-override** (**4**), **critical** (**5**), **internet** (**6**), **network** (**7**). • **tos**—Enter to match by type of service level, specified by a number from 0 to 15 or a name: **normal** (**0**), **max-reliability** (**2**), **max-throughput** (**4**), **min-delay** (**8**). • **dscp**—Enter to match packets with the DSCP value specified by a number from 0 to 63, or use the question mark (?) to see a list of available values.	**access-list** *access-list-number* **permit** *protocol* {*source source-wildcard destination destination-wildcard*} [**precedence** *precedence*] [**tos** *tos*] [**dscp** *dscp*] **Note** If you enter a **dscp** value, you cannot enter **tos** or **precedence**. You can enter both a **tos** and a **precedence** value with no **dscp**.

	Step	Command
Step 3	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*} [**precedence** *precedence*] [**tos** *tos*] [**dscp** *dscp*] as defined in Step 2.	**ip access-list extended** *name*
Step 4	Return to privileged EXEC mode.	**end**
Step 5	Verify your entries.	**show access-lists**

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

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

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

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

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

Creating Layer 2 MAC ACLs

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

	Step	Command
Step 1	Enter global configuration mode.	**configure terminal**
Step 2	Create a Layer 2 MAC ACL by specifying the name of the list and enter extended MAC ACL configuration mode.	**mac access-list extended** *name*

	Step	Command
Step 3	Always use the **permit** keyword for ACLs used as match criteria in QoS policies. • For *src-MAC-addr*, enter the MAC address of the host from which the packet is being sent. You can specify in hexadecimal format (H.H.H), use the **any** keyword for *source* 0.0.0, *source-wildcard* ffff.ffff.ffff, or use the **host** keyword for *source* 0.0.0. • For *mask,* enter the wildcard bits by placing ones in the bit positions that you want to ignore. • For *dst-MAC-addr*, enter the MAC address of the host to which the packet is being sent. You can specify in hexadecimal format (H.H.H), use the **any** keyword for *source* 0.0.0, *source-wildcard* ffff.ffff.ffff, or use the **host** keyword for *source* 0.0.0. • (Optional) For *type mask*, specify the Ethertype number of a packet with Ethernet II or SNAP encapsulation to identify the protocol of the packet. For *type*, the range is from 0 to 65535, typically specified in hexadecimal. For *mask*, enter the *don't care* bits applied to the Ethertype before testing for a match.	**permit** {**host** *src-MAC-addr mask* | **any** | **host** *dst-MAC-addr* | *dst-MAC-addr mask*} [*type mask*]
Step 4	Return to privileged EXEC mode.	**end**
Step 5	Verify your entries.	**show access-lists** [*access-list-number* | *access-list-name*]
Step 6	(Optional) Save your entries in the configuration file.	**copy running-config startup-config**

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

This example shows how to create a Layer 2 MAC ACL with two **permit** statements. The first statement allows traffic from the host with MAC address 0001.0000.0001 to the host with MAC address 0002.0000.0001. The second statement allows only Ethertype XNS-IDP traffic from the host with MAC address 0001.0000.0002 to the host with MAC address 0002.0000.0002:

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

Using Class Maps to Define a Traffic Class

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

Follow these guidelines when configuring class maps:

- A **match-all** class map cannot have more than one classification criterion (one match statement), but a **match-any** class map can contain multiple match statements.
- The **match cos** and **match vlan** commands are supported only on Layer 2 802.1Q trunk ports.
- You use a class map with the **match vlan** command in the parent policy in input hierarchical policy maps for per-port, per-VLAN QoS on trunk ports. A policy is considered a parent policy map when it has one or more of its classes associated with a child policy map. Each class within a parent policy map is called a parent class. You can configure only the **match vlan** command in parent classes. You cannot configure the **match vlan** command in classes within the child policy map.
- For an input policy map, you cannot configure an IP classification (**match ip dscp**, **match ip precedence**, **match access-group** for an IP ACL) and a non-IP classification (**match cos** or **match access-group** for a MAC ACL) in the same policy map or class map. For a per-port, per-VLAN hierarchical policy map, this applies to the child policy map.
- You cannot configure **match qos-group** for an input policy map.
- In an output policy map, no two class maps can have the same classification criteria; that is, the same match qualifiers and values.
- The maximum number of class maps on the switch module is 1024.

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

	Step	Command
Step 1	Enter global configuration mode.	**configure terminal**
Step 2	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).	**class-map** [**match-all** | **match-any**] *class-map-name*

	Step	Command
Step 3	Define the match criterion to classify traffic. By default, no match criterion is defined. Only one match type per class map is supported, and only one ACL per class map is supported. • For **access-group** *acl-index-or-name,* specify the number or name of an ACL. Matching access groups is supported only in input policy maps. • For **cos** *cos-list*, enter a list of up to four CoS values in a single line to match against incoming packets. Separate each value with a space. You can enter multiple *cos-list* lines to match more than four CoS values. The range is 0 to 7. • For **ip dscp** *dscp-list*, enter a list of up to eight IPv4 DSCP values to match against incoming packets. Separate each value with a space. You can enter multiple *dscp-list* lines to match more than eight DSCP values. The numerical range is 0 to 63. You can also configure DSCP values in other forms. See the "Classification Based on IP DSCP" section on page 14-8. • 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 **vlan** *vlan-list,* specify a VLAN ID or a range of VLANs to be used in a parent policy map for per-port, per-VLAN QoS on a trunk port. The VLAN ID range is 1 to 4094. • 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.	**match** {**access-group** *acl-index-or-name* | **cos** *cos-list* | **ip dscp** *dscp-list* | **ip precedence** *ip-precedence-list* | **qos-group** *value* | **vlan** *vlan-list*}
Step 4	Return to privileged EXEC mode.	**end**
Step 5	Verify your entries.	**show class-map**
Step 6	(Optional) Save your entries in the configuration file.	**copy running-config startup-config**

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 parent 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
```

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. You can specify table maps in **set** commands and use them as mark-down mapping for the policers.

These table maps are supported on the switch module:

- 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 module 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 for **violate-action** for aggregate policing unless you configure a table map for **exceed-action** and no explicit action is configured for **violate-action**.

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

	Step	Command
Step 1	Enter global configuration mode.	**configure terminal**
Step 2	Create a table map by entering a table-map name and entering table-map configuration mode.	**table-map** *table-map-name*
Step 3	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.	**map from** *from-value* **to** *to-value*
Step 4	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 module does not change the CoS value of unmapped DSCP values.	**default** {*default-value* | **copy** | **ignore**}
Step 5	Return to privileged EXEC mode.	**end**
Step 6	Verify your entries.	**show table-map** [*table-map-name*]
Step 7	(Optional) Save your entries in the configuration file.	**copy running-config startup-config**

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
```

Attaching a Traffic Policy to an Interface

You use the **service-policy** interface configuration command to attach a traffic policy to an interface and to specify the direction in which the policy should be applied: either an input policy map for incoming traffic or an output policy map for outgoing traffic. Input and output policy maps support different QoS features. See the "Configuring Input Policy Maps" section on page 14-47 and the "Configuring Output Policy Maps" section on page 14-66 for restrictions on input and output policy maps.

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

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

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

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

	Step	Command
Step 1	Enter global configuration mode.	**configure terminal**
Step 2	Specify the port to attach to the policy map, and enter interface configuration mode. Valid interfaces are physical ports.	**interface** *interface-id*
Step 3	Specify the policy-map name and whether it is an input policy map or an output policy map.	**service-policy** {**input** | **output**} *policy-map-name*
Step 4	Return to privileged EXEC mode.	**end**
Step 5	Verify your entries.	**show policy-map interface** [*interface-id*]
Step 6	(Optional) Save your entries in the configuration file.	**copy running-config startup-config**

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

Configuring Input Policy Maps

Policy maps specify which traffic class to act on and what actions to take. All traffic that fails to meet matching criteria of a traffic class belongs to the default class. Input policy maps regulate traffic entering the switch module. In an input policy, you can match CoS, DSCP, IP precedence, ACLs, or VLAN IDs and configure individual policing, aggregate policing, or marking to a CoS, DSCP, IP precedence, or QoS group value.

Follow these guidelines when configuring input policy maps:

- You can attach only one input policy map per port.

- The maximum number of policy maps configured on the switch module is 256.
- The number of configurable policer profiles on the switch module is 256. The number of supported policer instances on the switch module is 1,024 minus 1 more than the number of interfaces on the switch module.

  The number of policer instances for the Copper model (GRWIC-D-2S-8PC), which has 10 interfaces, is 1,013; the number of policer instances for the SFP model (GRWIC-D-6S), which has 6 interfaces, is 1,017.

  You can use a policer profile in multiple instances.
- The maximum number of classes in each input policy map is 64 plus **class-default**.
- The number of input policy maps that can be attached in a switch module is limited by the availability of hardware resources. If you attempt to attach an input policy map that causes any hardware resource limitation to be exceeded, the configuration fails.
- After you have attached a single-level policy map to an interface by using the **service-policy input** interface configuration command, you can modify the policy without detaching it from the interface. You can add or delete classification criteria, add or delete classes, add or delete actions, or change the parameters of the configured actions (policers, rates, mapping, marking, and so on). This also applies to changing criteria for the child policy of a hierarchical policy map, as in a per-port per-VLAN hierarchical policy map.

  For the parent policy of a hierarchical policy map, you cannot add or delete a class at the parent level if the policy map is attached to an interface. You must detach the policy from the interface, modify the policy, and then re-attach it to the interface.
- You can configure a maximum 2-level hierarchical policy map as an input policy map only with VLAN-based classification at the parent level and no VLAN-based classification at the child level.
- When an input policy map with only Layer 2 classification is attached to a routed port or a switch module port containing a routed SVI, the service policy acts only on switching eligible traffic and not on routing eligible traffic.
- On an 802.1Q tunnel port, you can use only an input policy map with Layer 2 classification based on MAC ACLs to classify traffic. Input policy maps with Layer 3 classification or with Layer 2 classification based on CoS or VLAN ID are not supported on tunnel ports.
- Input policy maps support policing and marking, not scheduling or queuing. You cannot configure **bandwidth**, **priority**, **queue-limit**, or **shape average** in input policy maps.

These sections describe how to configure different types of input policy maps:

- Configuring Input Policy Maps with Individual Policing, page 14-48
- Configuring Input Policy Maps with Aggregate Policing, page 14-54
- Configuring Input Policy Maps with Marking, page 14-58
- Configuring Per-Port Per-VLAN QoS with Hierarchical Input Policy Maps, page 14-59

Configuring Input Policy Maps with Individual Policing

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.

Follow these guidelines when configuring individual policers:

- Policing is supported only on input policy maps.

- The switch module supports a maximum of 229 policers. (228 user-configurable policers and 1 policer reserved for internal use).
- When CPU protection is enabled (the default), you can configure 45 ingress policers per port. If you disable CPU protection by entering the **no policer cpu uni all** global configuration command and reloading the switch module, you can configure a maximum of 63 policers per port for user-defined classes and one for class-default. You can enter the **show policer cpu uni-eni** {**drop** | **rate**} privileged EXEC command to see if CPU protection is enabled.
- When you use a table map for police exceed-action in an input policy map, the protocol type of the *map from* type of action must be the same as the protocol type of the associated classification. For example, if the associated class map represents an IP classification, the **map from** type of action that references the table map must be **dscp** or **precedence**. If the associated class map represents a non-IP classification, the **map from** type of action that references the table map must be **cos**.
- 2-rate, 3-color policing is supported only on input policy maps; 1-rate, 2-color policing is supported on both input and output policy maps.
- The number of policer instances on the switch module can be 1024 minus 1 more than the number interfaces. The switch module supports a maximum of 256 policer profiles.
- If you do not configure a violate-action, by default the violate class is assigned the same action as the exceed-action.

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
```

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

	Step	Command
Step 1	Enter global configuration mode.	**configure terminal**
Step 2	Create a policy map by entering the policy map name, and enter policy-map configuration mode. By default, no policy maps are defined.	**policy-map** *policy-map-name*
Step 3	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.	**class** {*class-map-name* | **class-default**}

	Step	Command
Step 4	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 (bps). The range is 8000 to 1000000000. • For **cir** *cir-bps,* specify a committed information rate at which the bc token bucket is updated in bits per second (b/s). The range is 8000 to 1000000000. • For *burst-bytes* (optional)*,* specify the normal burst size in bytes. The range is 8000 to 1000000. • (Optional) For **bc** *conform-burst,* specify the conformed burst used by the bc token bucket for policing.The range is 8000 to 1000000 bytes. • (Optional) For **pir** *pir-bps,* specify the peak information rate at which the be token bucket for policing is updated. The range is 8000 to 1000000000 b/s. If you do not enter a **pir** *pir-bps*, the policer is configured as a 1-rate, 2-color policer. • For **be** *peak-burst*, specify the peak burst size used by the be token bucket. The range is 8000 to 1000000 bytes. The default is internally calculated based on the user configuration.	**police** {*rate-bps* | **cir** {*cir-bps*} [*burst-bytes*] [**bc** [*conform-burst*] [**pir** *pir-bps* [**be** *peak-burst*]]

	Step	Command
Step 5	(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. The default is **drop**. • (Optional) For *action*, specify one of these actions to perform on the packets: – **drop**—Drop the packet. **Notes:** 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*—Enter a new CoS value to be assigned to the packet, and send the packet. The range is from 0 to 7. • **set-dscp-transmit** *dscp-value*—Enter a new IP DSCP value to be assigned to the packet, and send the packet. The range is from 0 to 63. You can also enter a mnemonic name for a commonly used value. • **set-prec-transmit** *cos-value*—Enter a new IP precedence value to be assigned to the packet, and send the packet. The range is from 0 to 7. • **set-qos-transmit** *qos-group-value*—Identify a qos-group to be used at egress to specify packets. The range is from 0 to 99. • **transmit**—Send the packet without altering it. You can enter a single conform-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.	**conform-action** [**drop** | **set-cos-transmit** {*cos_value* | [**cos** | **dscp** | **precedence**] [**table** *table-map name*]} | **set-dscp-transmit** {*dscp_value* | [**cos** | **dscp** | **precedence**] [**table** *table-map name*]} | **set-prec-transmit** {*precedence_value* | [**cos** | **dscp** | **precedence**] [**table** *table-map name*]} | **set-qos-transmit** *qos-group_value* | **transmit**] | **exceed-action** [**drop** | **set-cos-transmit** {*cos_value* | [**cos** | **dscp** | **precedence**] [**table** *table-map name*]} | **set-dscp-transmit** {*dscp_value* | [**cos** | **dscp** | **precedence**] [**table** *table-map name*]} | **set-prec-transmit** {*precedence_value* | [**cos** | **dscp** | **precedence**] [**table** *table-map name*]} | **set-qos-transmit** *qos-group_value* | **transmit**] | **violate- action** [**drop** | **set-cos-transmit** {*cos_value* | [**cos** | **dscp** | **precedence**] [**table** *table-map name*]} | **set-dscp-transmit** {*dscp_value* | [**cos** | **dscp** | **precedence**] [**table** *table-map name*]} | **set-prec-transmit** {*precedence_value* | [**cos** | **dscp** | **precedence**] [**table** *table-map name*]} | **set-qos-transmit** *qos-group_value* | **transmit**]
Step 6	Return to policy-map configuration mode.	**exit**
Step 7	Return to global configuration mode.	**exit**

	Step	Command
Step 8	Enter interface configuration mode for the interface to which you want to attach the policy.	**interface** *interface-id*
Step 9	Attach the policy map (created in Step 2) to the ingress interface.	**service-policy input** *policy-map-name*
Step 10	Return to privileged EXEC mode.	**end**
Step 11	Verify your entries.	**show policy-map** [*policy-map-name*| **interface**]
Step 12	(Optional) Save your entries in the configuration file.	**copy running-config startup-config**

After you have created an input policy map, you attach it to an interface in the input direction. See the "Attaching a Traffic Policy to an Interface" section on page 14-47.

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 fastethernet0/1
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
```

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 fastethernet0/1
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:

```
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 fastethernet0/1
Switch(config-if)# service-policy input in-policy
Switch(config-if)# 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 (bps) 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 fastethernet0/1
Switch(config-if)# service-policy input map1
Switch(config-if)# exit
```

This example shows how to use policy-map class police configuration mode to set exceed action mark-down using table-maps. The policy map sets a committed information rate of 23000 bps and a conform burst-size of 10000 bytes. The policy map includes the default conform action (**transmit**) and the exceed action to mark the Layer 2 CoS value based on the table map and to mark IP DSCP to af41:

```
Switch(config)# policy-map in-policy
Switch(config-pmap)# class in-class-1
Switch(config-pmap-c)# police cir 23000 bc 10000
Switch(config-pmap-c-police)# exceed-action set-cos-transmit cos table
police-cos-markdn-tablemap
Switch(config-pmap-c-police)# exceed-action set-dscp-transmit af41
Switch(config-pmap-c-police)# exit
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface fastethernet0/1
Switch(config-if)# service-policy input in-policy
Switch(config-if)# exit
```

Configuring Input Policy Maps with Aggregate Policing

You use the **policer aggregate** global configuration command to configure an aggregate policer. An aggregate policer is shared by multiple traffic classes within the same policy map. You define the aggregate policer, create a policy map, associate a class map with the policy map, associate the policy map with the aggregate policer, and apply the service policy to a port.

Follow these guidelines when configuring aggregate policers:

- Aggregate policing is supported only on input policy maps.
- The switch module supports a maximum of 229 policers associated with ports (228 user-configurable policers and 1 policer reserved for internal use). You can configure up to 45 policers on a port.
- When CPU protection is enabled (the default), you can configure 45 ingress policers per port. If you disable CPU protection by entering the **no policer cpu uni all** global configuration command and reloading the switch module, you can configure a maximum of 63 policers per port for user-defined classes and one for class-default. You can enter the **show policer cpu uni-eni** {**drop** | **rate**} privileged EXEC command to see if CPU protection is enabled.
- The maximum number of configured aggregate policers is 256.
- The number of policer instances on the switch module can be 1024 minus 1 more than the total number interfaces on the switch module. The switch module supports a maximum of 256 policer profiles.
- If you do not configure a violate-action, by default the violate class is assigned the same action as the exceed-action.
- Only one policy map can use any specific aggregate policer. Aggregate policing cannot be used to aggregate streams across multiple interfaces. You can use aggregate policing only to aggregate streams across multiple classes in a policy map attached to an interface and to aggregate traffic streams across VLANs on a port in a per-port, per-VLAN policy map.
- When you use a table map for police exceed-action in an input policy map, the protocol type of the map from type of action must be the same as the protocol type of the associated classification. For example, if the associated class map represents an IP classification, the **map from** type of action that references the table map must be either **dscp** or **precedence**. If the associated class map represents a non-IP classification, the **map from** type of action that references the table map must be **cos**.
- Table maps are not supported for violate-action for aggregate policing unless a table map is configured for **exceed-action** and no explicit action is configured for **violate-action**.

You can configure multiple conform, exceed, and violate actions simultaneously for an aggregate policer as parameters in the **policer aggregate** global configuration command, but you must enter the actions in this order:

- **conform-action** must be followed by **drop** or **transmit** or by **set** actions in this order:

  set-qos-transmit

  set-dscp-transmit or **set-prec-transmit**

  set-cos-transmit

- **exceed-action** must be followed by **drop** or **transmit** or by **set** actions in this order:

  set-qos-transmit

  set-dscp-transmit or **set-prec-transmit**

  set-cos-transmit

- **violate-action** must be followed by **drop** or **transmit** or by **set** actions in this order:
  - **set-qos-transmit**
  - **set-dscp-transmit or set-prec-transmit**
  - **set-cos-transmit**

Note You do not configure aggregate policer conform-action, exceed-action, and violate-action in policy-map class police configuration mode—you must enter all actions in a string. Consequently, if you enter multiple conform, exceed, and violate actions, the command can become quite long, in which case it might be truncated and difficult to read.

Beginning in privileged EXEC mode, follow these steps to create a 2-rate, 3-color aggregate policer:

	Step	Command
Step 1	Enter global configuration mode.	**configure terminal**
Step 2	Define the policer parameters that can be applied to multiple traffic classes within the same policy map. • For *aggregate-policer-name*, specify the name of the aggregate policer. • For *rate-bps,* specify average traffic rate in bits per second (b/s). The range is 8000 to 1000000000. • For **cir** *cir-bps,* specify a committed information rate (CIR) at which the first token bucket is updated in bits per second (b/s). The range is 8000 to 1000000000. • For *burst-bytes* (optional)*,* specify the normal burst size in bytes. The range is 8000 to 1000000. • (Optional) For **bc** *conform-burst,* specify the conformed burst used by the first token bucket for policing.The range is 8000 to 1000000 bytes.	**policer aggregate** *aggregate-policer-name* {*rate-bps* | **cir** *cir-bps*} [*burst-bytes*] [**bc** [*conform-burst*] [**pir** *pir-bps* [**be** *peak-burst*]] [**conform-action** [**drop** | **set-cos-transmit** {*cos_value* | [**cos** | **dscp** | **precedence**] [**table** *table-map name*]} | **set-dscp-transmit** {*dscp_value* | [**cos** | **dscp** | **precedence**] [**table** *table-map name*]} | **set-prec-transmit** {*precedence_value* | [**cos** | **dscp** | **precedence**] [**table** *table-map name*]} | **set-qos-transmit** *qos-group_value* | **transmit**] [**exceed-action** [**drop** | **set-cos-transmit** {*cos_value* | [**cos** | **dscp** | **precedence**] [**table** *table-map name*]} | **set-dscp-transmit** {*dscp_value* | [**cos** | **dscp** | **precedence**] [**table** *table-map name*]} | **set-prec-transmit** {*precedence_value* | [**cos** | **dscp** | **precedence**] [**table** *table-map name*]} | **set-qos-transmit** *qos-group_value* | **transmit**]] [**violate- action** [**drop** | **set-cos-transmit** {*cos_value* | [**cos** | **dscp** | **precedence**]} | **set-dscp-transmit** {*dscp_value* | [**cos** | **dscp** | **precedence**]} | **set-prec-transmit** {*precedence_value* | [**cos** | **dscp** | **precedence**]} | **set-qos-transmit** *qos-group_value* | **transmit**]]

	Step	Command
	• (Optional) For **pir** *pir-bps,* specify the peak information rate at which the second token bucket for policing is updated. The range is 8000 to 1000000000 bits per second. If you do not enter a **pir** *pir-bps*, the policer is configured as a 1-rate, 2-color policer. • For **be** *peak-burst*, specify the peak burst size used by the second token bucket. The range is 8000 to 1000000 bytes. The default is internally calculated based on the user configuration. • (Optional) For **conform-action**, specify the action to take on packets that conform to the CIR. The default is to send 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**. • (Optional) For **exceed-action**, specify the action to take on packets that exceed the CIR. The default is to drop the packet. • (Optional) For **violate-action**, specify the action to take on packets that exceed the CIR. The default is to drop the packet. For definitions of the action keywords, see the command reference for this release or the "Configuring Input Policy Maps with Individual Policing" section on page 14-48. **Note** You cannot configure table maps for **violate-action** for aggregate policing unless a table map is configured for **exceed-action** and no explicit action is configured for violate-action.	
Step 3	Create a policy map by entering the policy map name, and enter policy-map configuration mode.	**policy-map** *policy-map-name*
Step 4	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.	**class** {*class-map-name* | **class-default**}
Step 5	Apply an aggregate policer to multiple classes in the same policy map. For *aggregate-policer-name*, enter the name specified in Step 2.	**police aggregate** *aggregate-policer-name*

	Step	Command
Step 6	Return to policy-map configuration mode.	**exit**
Step 7	Return to global configuration mode.	**exit**
Step 8	Enter interface configuration mode for the interface to which you want to attach the policy.	**interface** *interface-id*
Step 9	Attach the policy map (created in Step 3) to the ingress interface.	**service-policy input** *policy-map-name*
Step 10	Return to privileged EXEC mode.	**end**
Step 11	Verify your entries.	**show policer aggregate** [*aggregate-policer-name*]
Step 12	(Optional) Save your entries in the configuration file.	**copy running-config startup-config**

After you have created an aggregate policer, you attach it to an ingress port. See the "Attaching a Traffic Policy to an Interface" section on page 14-47.

To remove the specified aggregate policer from a policy map, use the **no police aggregate** *aggregate-policer-name* policy map configuration mode. To delete an aggregate policer and its parameters, use the **no policer aggregate** *aggregate-policer-name* global configuration command.

This example shows how to create an aggregate policer and attach it to multiple classes within a policy map. The policy map is attached to an ingress port:

```
Switch(config)# policer aggregate example 10900000 80000 conform-action transmit
exceed-action drop
Switch(config)# class-map testclass1
Switch(config-cmap)# match access-group 1
Switch(config-cmap)# exit
Switch(config)# class-map testclass2
Switch(config-cmap)# match access-group 2
Switch(config-cmap)# exit
Switch(config)# policy-map testexample
Switch(config-pmap)# class testclass
Switch(config-pmap-c)# police aggregate example
Switch(config-pmap-c)# exit
Switch(config-pmap)# class testclass2
Switch(config-pmap-c)# police aggregate example
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface fastethernet0/1
Switch(config-if)# service-policy input testexample
Switch(config-if)# exit
```

This example shows how to create a 2-rate, 3-color aggregate policer and attach it to multiple classes within a policy map. The policy map is attached to an ingress port:

```
Switch(config)# policer aggregate example cir 10900000 pir 80000000 conform-action
transmit exceed-action drop violate-action drop
Switch(config)# class-map testclass1
Switch(config-cmap)# match access-group 1
Switch(config-cmap)# exit
Switch(config)# class-map testclass2
Switch(config-cmap)# match access-group 2
Switch(config-cmap)# exit
Switch(config)# policy-map testexample
Switch(config-pmap)# class testclass
Switch(config-pmap-c)# police aggregate example
Switch(config-pmap-c)# exit
Switch(config-pmap)# class testclass2
Switch(config-pmap-c)# police aggregate example
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface fastethernet0/1
Switch(config-if)# service-policy input testexample
Switch(config-if)# exit
```

Configuring Input Policy Maps with 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 marking in policy maps:

- You can configure a maximum of 100 QoS groups on the switch module.
- When you use a table map for marking in an input policy map, the protocol type of the map from type of action must be the same as the protocol type of the associated classification. For example, if the associated class map represents an IP classification, the **map from** type of action that references the table map must be either **dscp** or **precedence**. If the associated class map represents a non-IP classification, the **map from** type of action that references the table map must be **cos**.

Beginning in privileged EXEC mode, follow these steps to create an input policy map that marks traffic:

	Step	Command
Step 1	Enter global configuration mode.	**configure terminal**
Step 2	Create a policy map by entering the policy map name, and enter policy-map configuration mode.	**policy-map** *policy-map-name*
Step 3	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.	**class** {*class-map-name* | **class-default**}
Step 4	Mark traffic by setting a new value in the packet, specifying a table map, or specifying a QoS group. • For **qos-group** *value,* identify a QoS group to be used at egress to identify specific packets. The range is from 0 to 99. • 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. • For [**ip**] **precedence** *new-precedence,* enter a new IP-precedence value to be assigned to the classified traffic. The range is 0 to 7.	**set qos-group** *value* and/or **set cos** {*cos_value* | **cos** [**table** *table-map-name*] | **dscp** [**table** *table-map-name*] | **precedence** [**table** *table-map-name*]} and/or **set** [**ip**] **dscp** {*dscp_value* | **cos** [**table** *table-map-name*] | **dscp** [**table** *table-map-name*] | **precedence** [**table** *table-map-name*]} and/or **set** [**ip**] **precedence** {*precedence_value* | **cos** [**table** *table-map-name*] | **dscp** [**table** *table-map-name*] | **precedence** [**table** *table-map-name*]}

	Step	Command
	- You can also configure a CoS, DSCP, or IP precedence table and optionally enter the table name. If you do not enter **table** *table-map name*, the table map default behavior is **copy**. See the "Configuring Table Maps" section on page 14-45.	
Step 5	Return to policy-map configuration mode.	**exit**
Step 6	Return to global configuration mode.	**exit**
Step 7	Enter interface configuration mode for the interface to which you want to attach the policy.	**interface** *interface-id*
Step 8	Attach the policy map (created in Step 2) to the ingress interface.	**service-policy input** *policy-map-name*
Step 9	Return to privileged EXEC mode.	**end**
Step 10	Verify your entries.	**show policy-map** [*policy-map-name* [**class** *class-map-name*]]
Step 11	(Optional) Save your entries in the configuration file.	**copy running-config startup-config**

Use the **no** form of the appropriate command to delete a policy map or table map or remove an assigned CoS, DSCP, 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 fastethernet0/1
Switch(config-if)# service-policy input Example
Switch(config-if)# exit
```

Configuring Per-Port Per-VLAN QoS with Hierarchical Input Policy Maps

Per-port, per-VLAN QoS allows classification based on VLAN IDs for applying QoS for frames received on a given interface and VLAN. This is achieved by using a hierarchical policy map, with a parent policy and a child policy.

Note these guidelines and limitations when configuring per-port, per-VLAN QoS:

- The feature is supported only by using a two-level hierarchical input policy map, where the parent level defines the VLAN-based classification, and the child level defines the QoS policy to be applied to the corresponding VLAN or VLANs.

- You can configure multiple service classes at the parent level to match different combinations of VLANs, and you can apply independent QoS policies to each parent-service class using any child policy map
- A policy is considered a parent policy map when it has one or more of its classes associated with a child policy map. Each class within a parent policy-map is called a parent-class. In parent classes, you can configure only the **match vlan** class-map configuration command. You cannot configure the **match vlan** command in classes within the child policy map.
- A per-port, per-VLAN parent level class map supports only a child-policy association; it does not allow any actions to be configured. For a parent-level class map, you cannot configure an action or a child-policy association for the class **class-default**.
- You cannot configure a mixture of Layer 2 and Layer 3 class maps in a child policy map. When you attempt to associate such a child policy map with a parent policy, the configuration is rejected. However, you can associate Layer 2 child policies and Layer 3 child policies with different parent-level class maps.
- Per-port, per-VLAN QoS is supported only on 802.1Q trunk ports.
- When the child policy-map attached to a VLAN or set of VLANs contains only Layer 3 classification (**match ip dscp**, **match ip precedence**, **match IP ACLs**), take care to ensure that these VLANs are not carried on any other port besides the one on which the per-port, per-vlan policy is attached. Not following this rule could result in improper QoS behavior for traffic ingressing the switch module on these VLANs.
- We also recommend that you restrict VLAN membership on the trunk ports to which the per-port, per-VLAN is applied by using the **switchport trunk allowed vlan** interface configuration command. Overlapping VLAN membership between trunk ports that have per-port, per-VLAN policies with Layer 3 classification could also result in unexpected QoS behavior.

Configuring per-port, per-VLAN QoS includes these tasks:

- Creating Child-Policy Class Maps, page 14-61
- Creating Parent-Policy Class Maps, page 14-63
- Creating Child Policy Maps, page 14-63
- Creating a Parent Policy Map, page 14-64
- Attaching a Parent Policy Map to an Interface, page 14-64

Creating Child-Policy Class Maps

Beginning in privileged EXEC mode, follow these steps to create one or more child-policy class maps:

	Step	Command
Step 1	Enter global configuration mode.	**configure terminal**
Step 2	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).	**class-map** [**match-all** | **match-any**] *child-class-map-name*

	Step	Command
Step 3	Define the match criterion to classify traffic. By default, no match criterion is defined. Only one match type per class map is supported, and only one ACL per class map is supported. • For **access-group** *acl-index-or-name,* specify the number or name of an ACL. Matching access groups is supported only in input policy maps. • For **cos** *cos-list*, enter a list of up to four CoS values in a single line to match against incoming packets. Separate each value with a space. You can enter multiple *cos-list* lines to match more than four CoS values. The range is 0 to 7. • For **ip dscp** *dscp-list*, enter a list of up to eight IPv4 DSCP values to match against incoming packets. Separate each value with a space. You can enter multiple *dscp-list* lines to match more than eight DSCP values. The numerical range is 0 to 63. You can also configure DSCP values in other forms. See the "Classification Based on IP DSCP" section on page 14-8. • 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 **qos-group** *value,* specify the QoS group number. The range is 0 to99. Matching of QoS groups is supported only in output policy maps. • For **vlan** *vlan-list,* specify a VLAN ID or a range of VLANs to be used in a parent policy map for per-port, per-VLAN QoS on a trunk port. The VLAN ID range is 1 to 4094.	**match** {**access-group** *acl-index-or-name* | **cos** *cos-list* | **ip dscp** *dscp-list* | **ip precedence** *ip-precedence-list* | **qos-group** *value* | **vlan** *vlan-list*}
Step 4	Return to privileged EXEC mode.	**end**
Step 5	Verify your entries.	**show class-map**
Step 6	(Optional) Save your entries in the configuration file.	**copy running-config startup-config**

Creating Parent-Policy Class Maps

Beginning in privileged EXEC mode, follow these steps to create one or more parent-policy class maps:

	Step	Command
Step 1	Enter global configuration mode.	**configure terminal**
Step 2	Create a **match-any** class map for the parent policy, and enter class-map configuration mode. **Note** You can enter **match-all** or not enter either **match-any** or **match-all** (to default to **match-all**) if you are going to match only one VLAN ID.	**class-map match-any** *parent-class-map-name*
Step 3	Define the VLAN or VLANs on which to classify traffic. For *vlan-id,* specify a VLAN ID, a series of VLAN IDs separated by a space, or a range of VLANs separated by a hyphen to be used in a parent policy map for per-port, per-VLAN QoS on a trunk port. The VLAN ID range is 1 to 4094. You can also enter the **match vlan** command multiple times to match multiple VLANs.	**match vlan** *vlan-id*
Step 4	Return to privileged EXEC mode.	**end**
Step 5	Verify your entries.	**show class-map**
Step 6	(Optional) Save your entries in the configuration file.	**copy running-config startup-config**

Creating Child Policy Maps

Beginning in privileged EXEC mode, follow these steps to create one or more child policy maps:

	Step	Command
Step 1	Enter global configuration mode.	**configure terminal**
Step 2	Create a child policy map by entering the policy map name, and enter policy-map configuration mode.	**policy-map** *child-policy-map-name*
Step 3	Enter a child class-map name or **class-default** to match all unclassified packets, and enter policy-map class configuration mode.	**class** {*child-class-map-name* | **class-default**}
Step 4	Use the **police** policy-map class configuration command to configure policers and the action to take for a class of traffic, or use the **set** policy-map class configuration command to mark traffic belonging to the class.	
Step 5	Return to privileged EXEC mode.	**end**

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

Creating a Parent Policy Map

Beginning in privileged EXEC mode, follow these steps to create a parent policy map and attach it to an interface:

	Step	Command
Step 1	Enter global configuration mode.	**configure terminal**
Step 2	Create a parent policy map by entering the policy map name, and enter policy-map configuration mode.	**policy-map** *parent-policy-map-name*
Step 3	Enter the parent class-map name, and enter policy-map class configuration mode.	**class** *parent-class-map-name*
Step 4	Associate the child policy map with the parent policy map	**service policy** *child-policy-map-name*
Step 5	Return to privileged EXEC mode.	**end**
Step 6	Verify your entries.	**show policy-map** [*parent-policy-map-name* [**class** *class-map-name*]]
Step 7	(Optional) Save your entries in the configuration file.	**copy running-config startup-config**

Attaching a Parent Policy Map to an Interface

Beginning in privileged EXEC mode, follow these steps to create attach the parent policy map to an interface:

	Step	Command
Step 1	Enter global configuration mode.	**configure terminal**
Step 2	Enter interface configuration mode for the interface to which you want to attach the policy.	**interface** *interface-id*
Step 3	Configure the port as a trunk port.	**switchport mode trunk**
Step 4	(Recommended) Restrict VLAN membership for trunk ports to avoid overlapping VLAN membership if the per-port, per-VLAN policy includes Layer 3 classification.	**switchport trunk allowed vlan** *vlan-list*
Step 5	Attach the parent policy map (created in the previous section) to the ingress interface.	**service-policy input** *parent- policy-map-name*

	Step	Command
Step 6	Return to privileged EXEC mode.	**end**
Step 7	Verify your entries.	**show policy-map interface** [*interface-id*]
Step 8	(Optional) Save your entries in the configuration file.	**copy running-config startup-config**

This is an example of using multiple parent classes to classify matching criteria for voice and video on customer VLANs:

```
Switch(config)# class-map match-any video
Switch(config-cmap)# match ip dscp af41
Switch(config-cmap)# exit
Switch(config)# class-map match-any voice
Switch(config-cmap)# match ip dscp ef
Switch(config-cmap)# exit

Switch(config)# class-map match-any customer1-vlan
Switch(config-cmap)# match vlan 100-105
Switch(config-cmap)# exit
Switch(config)# class-map match-any customer2-vlan
Switch(config-cmap)# match vlan 110-120
Switch(config-cmap)# exit

Switch(config)# policy-map child-policy-1
Switch(config-pmap)# class voice
Switch(config-pmap-c)# police cir 10000000 bc 50000
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-c)# exit
Switch(config-pmap)# class video
Switch(config-pmap-c)# set cos 4
Switch(config-pmap-c)# exit
Switch(config-pmap)# class class-default
Switch(config-pmap-c)# set cos 0
Switch(config-pmap-c)# exit

Switch(config)# policy-map child-policy-2
Switch(config-pmap)# class voice
Switch(config-pmap-c)# police cir 5000000
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-c)# exit
Switch(config-pmap)# class video
Switch(config-pmap-c)# police cir 40000000
Switch(config-pmap-c-police)# conform-action set-cos-transmit 4
Switch(config-pmap-c-police)# exceed-action set-cos-transmit 1
Switch(config-pmap-c-police)# exit
Switch(config-pmap-c)# exit
Switch(config-pmap)# class class-default
Switch(config-pmap-c)# set cos 0
Switch(config-pmap-c)# exit

Switch(config)# policy-map uni-parent
Switch(config-pmap)# class customer1-vlan
Switch(config-pmap-c)# service-policy child-policy-1
Switch(config-pmap-c)# exit
Switch(config-pmap)# class customer2-vlan
Switch(config-pmap-c)# service-policy child-policy-2
Switch(config-pmap-c)# exit

Switch(config)# interface fastethernet0/1
Switch(config-if)# switchport mode trunk
Switch(config-if)# switchport trunk allowed vlan 100-105, 110-120
Switch(config-if)# service-policy input uni-parent
Switch(config-pmap-c)# exit
```

Configuring Output Policy Maps

You use output policy maps to manage congestion avoidance, queuing, and scheduling of packets leaving the switch module. The switch module has four egress queues, and you use output policy maps to control the queue traffic. You configure shaping, queue-limit, and bandwidth on these queues. You can use high priority (class-based priority queuing). Policing is not supported on output policy maps, except when configuring priority with police for class-based priority queuing. Output policy map classification criteria are matching a CoS, DSCP, or IP precedence value or a QoS group.

Follow these guidelines when configuring output policy maps on physical ports:

- You can configure and attach as many output policy maps as there are ports on the switch module. Multiple output policy maps can use the same queue-limit configuration. However, these policy maps can have only three unique queue-limit configurations.
- Output policy maps can have a maximum of four classes, including the class **class-default**.
- All output policy maps must have the same number of defined class-maps defined, either 1, 2, or 3.
- All output policy maps must use the same set of classes, although the actions for each class can differ for each output policy map.
- In a child policy map, the **class-default** supports all output policy map actions except **priority** and **police**. Action restrictions for **class-default** are the same as for other classes except that a queue limit configuration for **class-default** does not require a scheduling action.
- To classify based on criteria at the output, the criteria must be established at the input. You can establish criteria at the input through classification only when you configure only policing and not marking, or through explicit marking when you configure any marking (policing with **conform** or **exceed** marking or unconditional **set** marking).
- You cannot configure class-based priority queuing under the class **class-default** in an output policy map
- In an output policy map, unless priority queuing is configured, the class default receives a minimum bandwidth guarantee equal to the unconfigured bandwidth on the port.
- After you have attached an output policy map to an interface by using the **service-policy** interface configuration command, you can change only the parameters of the configured actions (rates, percentages, and so on) or add or delete classification criteria of the class map while the policy map is attached to the interface. To add or delete a class or action, you must detach the policy map from all interfaces, modify it, and then reattach it to interfaces.

Note If you anticipate that you might need three classes in a policy map, you should define three classes when you create the policy map, even if you are not ready to use all three at that time. You cannot add a class to a policy map after it has been attached to an interface.

- When at least one output policy map is attached to a active port, other active ports without output policy maps attached might incorrectly schedule and incorrectly order traffic that uses the same classes as the attached output policy maps. We recommend attaching output policy maps to all ports that are in use. We also recommend putting any unused ports in the shutdown state by entering the **shutdown** interface configuration command. For example, if you attach an output policy map that shapes DSCP 23 traffic to a port, DSCP traffic that is sent out of any other port without a policy map attached could be incorrectly scheduled or ordered incorrectly with respect to other traffic sent out of the same port.
- We strongly recommended that you disable port speed autonegotiation when you attach an output policy map to a port to prevent the port from autonegotiating to a rate that would make the output policy map invalid. You can configure a static port speed by using the **speed** interface configuration command. If an output policy-map is configured on a port that is set for autonegotiation and the speed autonegotiates to a value that invalidates the policy, the port is put in the error-disabled state.
- You can attach only one output policy map per port.
- The maximum number of policy maps configured on the switch module is 256.

These sections describe how to configure different types of output policy maps:

- Configuring Output Policy Maps with Class-Based-Weighted-Queuing, page 14-67
- Configuring Output Policy Maps with Class-Based Shaping, page 14-69
- Configuring Output Policy Maps with Port Shaping, page 14-70
- Configuring Output Policy Maps with Class-Based Priority Queuing, page 14-72
- Configuring Output Policy Maps with Weighted Tail Drop, page 14-77

Configuring Output Policy Maps with Class-Based-Weighted-Queuing

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

Follow these guidelines when configuring CBWFQ:

- When configuring bandwidth in a policy map, all rate configurations must be in the same format, either a configured rate or a percentage.
- The total rate of the minimum bandwidth guarantees for each queue of the policy cannot exceed the total speed for the interface.
- You cannot configure CBWFQ (**bandwidth**) and traffic (**shape average**) or priority queuing (**priority**) for the same class in an output policy map.
- You cannot configure bandwidth as an absolute rate or a percentage of total bandwidth when strict priority (priority without police) is configured for another class map.
- You can configure bandwidth as a percentage of remaining bandwidth only when strict priority (priority without police) is configured for another class in the output policy map.
- When you configure CIR bandwidth for a class as an absolute rate or a percentage of total bandwidth, any excess bandwidth that remains after servicing the CIR of all classes in the policy map is divided among the classes the same proportion as the CIR rates. If you configure the CIR rate of a class to be 0, that class is not eligible for any excess bandwidth and will receive no bandwidth.

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:

	Step	Command
Step 1	Enter global configuration mode.	**configure terminal**
Step 2	Create a policy map by entering the policy map name, and enter policy-map configuration mode.	**policy-map** *policy-map-name*
Step 3	Enter a *child class-map name* or **class-default** to match all unclassified packets, and enter policy-map class configuration mode.	**class** {*class-map-name* | **class-default**}
Step 4	Set output bandwidth limits for the policy-map class. • Enter a *rate* to set bandwidth in kilobits per second. The range is from 64 to 1000000. • Enter **percent** *value* to set bandwidth as a percentage of the total bandwidth. The range is 1 to 100 percent. • Enter **remaining percent** *value* to set bandwidth as a percentage of the remaining bandwidth. The range is 1 to 100 percent. This keyword is valid only when strict priority (priority without police) is configured for another class in the output policy map.	**bandwidth** {*rate* | **percent** *value* | **remaining percent** *value*}
	You must specify the same units in each bandwidth configuration in an output policy (absolute rates or percentages). The total guaranteed bandwidth cannot exceed the total available rate.	
Step 5	Return to policy-map configuration mode.	**exit**
Step 6	Return to global configuration mode.	**exit**
Step 7	Enter interface configuration mode for the interface to which you want to attach the policy.	**interface** *interface-id*
Step 8	Attach the policy map (created in Step 2) to the egress interface.	**service-policy output** *policy-map-name*
Step 9	Return to privileged EXEC mode.	**end**
Step 10	Verify your entries.	**show policy-map** [*policy-map-name* [**class** *class-map-name*]]
Step 11	(Optional) Save your entries in the configuration file.	**copy running-config startup-config**

After you have created an output policy map, you attach it to an egress port. See the "Attaching a Traffic Policy to an Interface" section on page 14-47.

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 set the precedence of a queue by allocating 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
```

Configuring Output Policy Maps with Class-Based Shaping

You use the **shape average** policy-map class configuration command to configure traffic shaping. Class-based shaping is a control mechanism that is applied to classes of traffic leaving an interface and uses the shape average command to limit the rate of data transmission used for the committed information rate (CIR) for the class.

Follow these guidelines when configuring class-based shaping:

- Configuring a queue for traffic shaping sets the maximum bandwidth or peak information rate (PIR) of the queue. In the switch module, configuring traffic shaping automatically also sets the minimum bandwidth guarantee or CIR of the queue to the same value as the PIR.
- You cannot configure CBWFQ (**bandwidth**) or priority queuing (**priority**) and traffic (**shape average**) for the same class in an output policy map.
- You cannot configure traffic shaping for a traffic class when strict priority (priority without police) is configured for another class within the output policy-map.

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:

	Step	Command
Step 1	Enter global configuration mode.	**configure terminal**
Step 2	Create a policy map by entering the policy map name, and enter policy-map configuration mode.	**policy-map** *policy-map-name*
Step 3	Enter a *child class-map name* or **class-default** to match all unclassified packets, and enter policy-map class configuration mode.	**class** {*class-map-name* | **class-default**}
Step 4	Specify the average class-based shaping rate. For *target bps*, specify the average bit rate in bits per second. The range is from 64000 to 1000000000.	**shape average** *target bps*

	Step	Command
Step 5	Return to policy-map configuration mode.	**exit**
Step 6	Return to global configuration mode.	**exit**
Step 7	Enter interface configuration mode for the interface to which you want to attach the policy.	**interface** *interface-id*
Step 8	Attach the policy map (created in Step 2) to the egress interface.	**service-policy output** *policy-map-name*
Step 9	Return to privileged EXEC mode.	**end**
Step 10	Verify your entries.	**show policy-map** [*policy-map-name* [**class** *class-map-name*]]
Step 11	(Optional) Save your entries in the configuration file.	**copy running-config startup-config**

After you have created an output policy map, you attach it to an egress port. See the "Attaching a Traffic Policy to an Interface" section on page 14-47.

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 Fast Ethernet port so that *outclass1*, *outclass2*, and *outclass3* get a maximum of 50, 20, and 10 Mb/s of the available port bandwidth. The class **class-default** gets the remaining 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 fastethernet0/1
Switch(config-if)# service-policy output out-policy
Switch(config-if)# exit
```

Configuring Output Policy Maps with 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 actions for the queues on the shaped port.

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

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

	Step	Command
Step 1	Enter global configuration mode.	**configure terminal**
Step 2	Create a hierarchical policy map by entering the hierarchical policy map name, and enter policy-map configuration mode for the parent policy.	**policy-map** *policy-map-name*
Step 3	Enter a policy-map class configuration mode for the default class.	**class class-default**
Step 4	Specify the average class-based shaping rate. For *target bps*, specify the average bit rate in bits per second. The range is from 4000000 to 1000000000.	**shape average** *target bps*
Step 5	Specify the child policy-map to be used in the hierarchical policy map if required.	**service-policy** *policy-map-name*
Step 6	Return to policy-map configuration mode.	**exit**
Step 7	Return to global configuration mode.	**exit**
Step 8	Enter interface configuration mode for the interface to which you want to attach the policy.	**interface** *interface-id*
Step 9	Attach the parent policy map (created in Step 2) to the egress interface.	**service-policy output** *policy-map-name*
Step 10	Return to privileged EXEC mode.	**end**
Step 11	Verify your entries.	**show policy-map** [*policy-map-name* [**class** *class-map-name*]]
Step 12	(Optional) Save your entries in the configuration file.	**copy running-config startup-config**

After you have created the hierarchical output policy map, you attach it to an egress port. See the "Attaching a Traffic Policy to an Interface" section on page 14-47.

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 Mbps, 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 fastethernet0/1
Switch(config-if)# service-policy output out-policy-parent
Switch(config-if)# exit
```

Configuring Output Policy Maps with 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. With strict priority queuing, the priority queue is constantly serviced; all packets in the queue are scheduled and sent until the queue is empty. Excessive use of the priority queues can possibly delay packets in other queues and create unnecessary congestion.

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

- You can associate the **priority** command with a single unique class for all attached output policies on the switch module.
- When you configure a traffic class as a priority queue, you can configure only **police** and **queue-limit** as other queuing actions for the same class. You cannot configure **bandwidth** or **shape average** with priority queues in the same class.
- You cannot associate the **priority** command with the **class-default** of the output policy map.

Configuring Priority Without Police

Follow these guidelines when configuring strict priority queuing (priority without police):

- You cannot configure priority queuing without policing for a traffic class when class-based shaping (**shape average**) or CBWFQ (**bandwidth**) is configured for another class within the output policy-map.
- When you configure priority queuing without policing for a traffic class, you can only configure the other queues for sharing by using the **bandwidth remaining percent** policy-map class configuration command to allocate excess bandwidth. This command does not guarantee the allocated bandwidth, but does ensure the rate of distribution.

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

	Step	Command
Step 1	Enter global configuration mode.	**configure terminal**
Step 2	Create classes for three egress queues. Enter match conditions classification for each class.	**class-map** *class-map-name*
Step 3	Create a policy map by entering the policy map name, and enter policy-map configuration mode.	**policy-map** *policy-map-name*
Step 4	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.	**class** *class-map-name*
Step 5	Set the strict scheduling priority for this class. **Note** Only one unique class map on the switch module can be associated with a **priority** command. You cannot configure priority along with any other queuing action (**bandwidth** or **shape average**).	**priority**
Step 6	Exit policy-map class configuration mode for the priority class.	**exit**

	Step	Command
Step 7	Enter the name of a nonpriority class, and enter policy-map class configuration mode for that class.	**class** *class-map-name*
Step 8	Set output bandwidth limits for the policy-map class as a percentage of the remaining bandwidth. The range is 1 to 100 percent.	**bandwidth remaining percent** *value*
Step 9	Exit policy-map class configuration mode for the class	**exit**
Step 10	Return to global configuration mode.	**exit**
Step 11	Enter interface configuration mode for the interface to which you want to attach the policy.	**interface** *interface-id*
Step 12	Attach the policy map (created in Step 3) to the egress interface.	**service-policy output** *policy-map-name*
Step 13	Return to privileged EXEC mode.	**end**
Step 14	Verify your entries.	**show policy-map**
Step 15	(Optional) Save your entries in the configuration file.	**copy running-config startup-config**

After you have created an output policy map, you attach it to an egress port. See the "Attaching a Traffic Policy to an Interface" section on page 14-47.

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 Priority With Police

You can use the priority with police feature and configure an unconditional priority policer to limit the bandwidth used by the priority queue and allocate bandwidth or shape other queues. Follow these guidelines when configuring priority with police:

- You cannot configure a policer committed burst size for an unconditional priority policer even though the keyword is visible in the CLI help. Any configured burst size is ignored when you try to attach the output service policy.
- The allowed police rate range is 64000 to 1000000000 bps, even though the range that appears in the CLI help is 8000 to 1000000000. You cannot attach an output service policy with an out-of-range rate.
- You cannot configure priority with policing for a traffic class when **bandwidth remaining percent** is configured for another class in the same output policy map.
- You can configure 1-rate, 2-color policers for output policies with priority. You cannot configure 2-rate, 3-color policers for output policies.

Beginning in privileged EXEC mode, follow these steps to configure priority with police:

	Step	Command
Step 1	Enter global configuration mode.	**configure terminal**
Step 2	Create classes for three egress queues. Enter match conditions classification for each class.	**class-map** *class-map-name*
Step 3	Create a policy map by entering the policy map name, and enter policy-map configuration mode.	**policy-map** *policy-map-name*
Step 4	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.	**class** *class-map-name*
Step 5	Configure this class as the priority class. **Note** Only one unique class map on the switch module can be associated with a **priority** command.	**priority**

	Step	Command
Step 6	Define a policer for the priority class of traffic. • For *rate-bps,* specify average traffic rate in bits per second (bps). The range is 64000 to 1000000000. **Note** When you use the **police** command with the **priority** command in an output policy, the police rate range and the CIR range is 64000 to 1000000000 bps, even though the range that appears in the CLI help is 8000 to 1000000000. You cannot attach an output service policy with an out-of-range rate. • For **cir** *cir-bps,* specify a committed information rate (CIR) in bits per second (bps). The range is 64000 to 1000000000. Although visible in the command-line help string, the burst-size option is not supported in output policy maps. You cannot attach an output service policy map that has a configured burst size.	**police** {*rate-bps* | **cir** *cir-bps*}
Step 7	(Optional) Enter the action to be taken on packets that conform to the CIR. If no action is entered, the default action is to send the packet. **Note** You can enter a single conform-action as part of the command string following the **police** command. You can also enter a carriage return after the **police** command and enter policy-map class police configuration mode to enter the conform-action. When the *police* command is configured with priority in an output policy map, only the default conform-action of **transmit** is supported. Although visible in the command-line help string, the other police conform actions are not supported in output policy maps.	**conform-action** [**transmit**]

	Step	Command
Step 8	(Optional) Enter the action to be taken for packets that do not conform to the CIR. If no action is entered, the default action is to drop the packet. **Note** You can enter a single exceed-action as part of the command string following the **police** command. You can also enter a carriage return after the **police** command and enter policy-map class police configuration mode to enter the exceed-action. When the *police* command is configured with priority in an output policy map, only the default exceed-action of **drop** is supported. Although visible in the command-line help string, the other police exceed actions are not supported in output policy maps.	**exceed-action** [**drop**]
Step 9	Exit policy-map class configuration mode for the priority class.	**exit**
Step 10	Enter the name of the first nonpriority class, and enter policy-map class configuration mode for that class.	**class** *class-map-name*
Step 11	Set output bandwidth limits for the policy-map class in kilobits per second (the range is 64 to 1000000) or a percentage of the total bandwidth (the range is 1 to 100 percent) or specify the average class-based shaping rate in bits per second (the range is 64000 to 1000000000).	**bandwidth** {**rate** | **percent** *value*} or **shape average** *target bps*
Step 12	Return to policy-map configuration mode.	**exit**
Step 13	Return to global configuration mode.	**exit**
Step 14	Enter interface configuration mode for the interface to which you want to attach the policy.	**interface** *interface-id*
Step 15	Attach the policy map (created in Step 3) to the egress interface.	**service-policy output** *policy-map-name*
Step 16	Return to privileged EXEC mode.	**end**
Step 17	Verify your entries.	**show policy-map**
Step 18	(Optional) Save your entries in the configuration file.	**copy running-config startup-config**

After you have created an output policy map, you attach it to an egress port. See the "Attaching a Traffic Policy to an Interface" section on page 14-47.

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

This example shows how to use the **priority** with **police** commands to configure *out-class1* as the priority queue, with traffic going to the queue limited to 20000000 bps so that the priority queue never uses more than that. Traffic above that rate is dropped. The other traffic queues are configured as in the previous example:

```
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 percent 50
Switch(config-pmap-c)# exit
Switch(config-pmap)# class out-class3
Switch(config-pmap-c)# bandwidth 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 Output Policy Maps with Weighted Tail Drop

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

Follow these guidelines when configuring WTD:

- Configuring WTD with the **queue-limit** command is supported only when you first configure a scheduling action, such as **bandwidth**, **shape average**, or **priority**. The exception to this is when you are configuring **queue-limit** in the **class-default**.
- You can configure and attach as many output policy maps as there are ports. Multiple output policy maps can use the same queue-limit configuration. However, these policy maps can have only three unique queue-limit configurations.
- You can use the **queue-limit** command to configure the queue-limit for CPU-generated traffic.
- When you use the **queue-limit** command to configure queue thresholds for a class, the WTD thresholds must be less than or equal to the queue maximum threshold. A queue size configured with no qualifier must be larger than any queue sizes configured with qualifiers.
- You cannot configure more than two unique threshold values for the WTD qualifiers (**cos**, **dscp**, **precedence**, or **qos-group**) in the **queue-limit** command. However, there is no limit to the number of qualifiers that you can map to those thresholds. You can configure a third unique threshold value to set the maximum queue, using the **queue-limit** command with no qualifiers.
- A WTD qualifier in the **queue-limit** command must be the same as at least one **match** qualifier in the associated class map.
- In an output policy map, when you configure a queue-limit for a unique class, all other output policy maps must use the same format of qualifier type and qualifier value. Only queue-limit threshold values can be different. For example, when you configure class A queue-limit thresholds for **dscp 30** and **dscp 50** in *policy-map1*, and you configure class A queue-limits in policy-map 2, you must use **dscp 30** and **dscp 50** as qualifiers. You cannot use **dscp 20** and **dscp 40**. The threshold values can be different, but different threshold values would create a new unique queue-limit configuration.

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

	Step	Command
Step 1	Enter global configuration mode.	**configure terminal**
Step 2	Create a policy map by entering the policy map name, and enter policy-map configuration mode.	**policy-map** *policy-map-name*
Step 3	Enter a child 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 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 skip Step 4.	**class** {*class-map-name* | **class-default**}
Step 4	Configure a scheduling action for the traffic class. For more information, see: • "Configuring Output Policy Maps with Class-Based-Weighted-Queuing" section on page 14-67 • "Configuring Output Policy Maps with Class-Based Shaping" section on page 14-69 • "Configuring Output Policy Maps with Port Shaping" section on page 14-70 • "Configuring Output Policy Maps with Class-Based Priority Queuing" section on page 14-72.	**bandwidth** {*rate* | **percent** *value* | **remaining percent** *value*} or **shape average** *target bps* or **priority**

	Step	Command
Step 5	Specify the queue size for the traffic class. • (Optional) For **cos** *value*, specify a CoS value. The range is from 0 to 7. • (Optional) For **dscp** *value*, specify a DSCP value. The range is from 0 to 63. • (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. • For *number-of-packets*, set the minimum threshold for WTD. The range is from 16 to 544, in multiples of 16, where each packet is a fixed unit of 256 bytes. **Note** For optimal performance, we strongly recommend that you configure the queue-limit to 272 or less. The value is specified in packets by default, but the **packets** keyword is optional. **Note** Multiple output policy maps can use the same queue-limit configuration. However these policy maps can have only three unique queue-limit configurations.	**queue-limit** [**cos** *value* | **dscp** *value* | **precedence** *value* | **qos-group** *value*] *number-of-packets* [**packets**]}
Step 6	Return to policy-map configuration mode.	**exit**
Step 7	Return to global configuration mode.	**exit**
Step 8	Enter interface configuration mode for the interface to which you want to attach the policy.	**interface** *interface-id*
Step 9	Attach the policy map (created in Step 2) to the egress interface. **Note** If you try to attach an output policy map that contains a fourth queue-limit configuration, you see an error message, and the attachment is not allowed.	**service-policy output** *policy-map-name*
Step 10	Return to privileged EXEC mode.	**end**
Step 11	Verify your entries.	**show policy-map** [*policy-map-name* [**class** *class-map-name*]]
Step 12	(Optional) Save your entries in the configuration file.	**copy running-config startup-config**

After you have created an output policy map, you attach it to an egress port. See the "Configuring Output Policy Maps" section on page 14-66.

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

This example shows a policy map with a specified bandwidth and queue size. Traffic that is not DSCP 30 or 10 is assigned a queue limit of 112 packets. Traffic with a DSCP value of 30 is assigned a queue-limit of 48 packets, and traffic with a DSCP value of 10 is assigned a queue limit of 32 packets. All traffic not belonging to the class traffic is classified into class-default, which is configured with 10 percent of the total available bandwidth and a large queue size of 256 packets:

```
Switch(config)# policy-map gold-policy
Switch(config-pmap)# class traffic
Switch(config-pmap-c)# bandwidth percent 50
Switch(config-pmap-c)# queue-limit 112
Switch(config-pmap-c)# queue-limit dscp 30 48
Switch(config-pmap-c)# queue-limit dscp 10 32
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config-pmap)# class class-default
Switch(config-pmap-c)# bandwidth percent 10
Switch(config-pmap-c)# queue-limit 256
Switch(config-pmap-c)# exit
Switch(config-pmap)# exit
Switch(config)# interface gigabitethernet0/1
Switch(config-if)# service-policy output gold-policy
Switch(config-if)# exit
```

Configuring QoS Marking and Queuing for CPU-Generated Traffic

Beginning in privileged EXEC mode, follow these steps to configure marking and queuing of CPU-generated traffic. This procedure is optional.

	Step	Command
Step 1	Enter global configuration mode.	**configure terminal**
Step 2	Refer to the "Configuring Table Maps" section on page 14-45.	Configure global table maps
Step 3	Mark traffic by setting a new CoS value or by specifying a table map. • For *cos-value*, enter a new CoS value. The range is from 0 to 7. • You can also mark CoS based on the CoS, DSCP, or IP-precedence value. You can optionally use a table map to configure CoS. If you do not enter **table-map** *table-map-name*, the table map default behavior is **copy**. See the "Table Maps" section on page 14-14. When you complete this step, go to Step 7.	**cpu traffic qos cos** {*cos-value* | **cos [table-map** *table-map-name*] | **dscp [table-map** *table-map-name*] | **prec [table-map** *table-map-name*]}

	Step	Command
Step 4	Mark traffic by setting a new DsCP value or by specifying a table map. - For **dscp** *new-dscp*, enter a new DSCP value for the classified traffic. The range is 0 to 63. - You can also configure a table map to mark DSCP based on the CoS, DSCP, or IP-precedence value. You can optionally enter the table name. If you do not enter **table-map** *table-map-map* name, the table map default behavior is **copy**. See the "Table Maps" section on page 14-14. - For additional DSCP classification options, see the "Classification Based on IP DSCP" section on page 14-8. When you complete this step, go to Step 7.	**cpu traffic qos dscp** {*dscp_value* | **cos [table-map** *table-map-name*] | **dscp [table-map** *table-map-name*] | **prec [table-map** *table-map-name*]}
Step 5	Mark traffic by setting a new precedence value or by specifying a table map. - For **precedence** *new-precedence*, enter a new IP-precedence value 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**). - You can also configure a table map to mark precedence based on the CoS, DSCP, or IP-precedence value. You can optionally enter the table name. If you do not enter **table-map** *table-map-map name*, the table map default behavior is **copy**. See the "Table Maps" section on page 14-14. - When you complete this step, go to Step 7.	**cpu traffic qos precedence** {*precedence_value* | **cos [table-map** *table-map-name*] | **dscp [table-map** *table-map-name*] | **prec** [**table-map** *table-map-name*]}
Step 6	Mark traffic by using a QoS group. For *qos-group-value*, identify a QoS group to use at egress. The range is 0 to 99. When you complete this step, go to Step 7.	**cpu traffic qos qos-group** *qos-group-value*
Step 7	Return to privileged EXEC mode.	**end**
Step 8	Refer to the "Configuring Output Policy Maps" section on page 14-66.	Configure output policy maps to map QoS markings like COS, IP DSCP, IP precedence and QoS group to class queues, configure queuing and scheduling
Step 9	(Optional) Save your entries in the configuration file.	**copy running-config startup-config**
Step 10	Display the configured class maps, policy maps, table maps, and CPU traffic QoS settings.	**show running-config**
Step 11	Display the QoS marking values for CPU-generated traffic.	**show cpu traffic qos**

	Step	Command
Step 12	Display information for all table maps or the specified table map.	**show table-map [***table-map-name]*
Step 13	Display QoS policy map information for the specified policy map name, interface, input or output policy maps, or policy-map class.	**show policy-map** [*policy-map-name* | **interface** [*interface-id*] [**output**] [**class** *class-name*]]

To disable any command, use the **no** form of the command.

Example 1

This example shows how to configure egress queuing based on the DSCP value of CPU-generated IP packets.

- All CPU-generated IP traffic queues on the egress port, based on its IP DSCP value, and the configured output policy map *output-policy*.
- All IP SLA or TWAMP probes with the DSCP value *ef* to simulate voice traffic are assigned to the *voice* class
- All IP SLA or TWAMP probes with the DSCP values *af41*, *af42* and *af43* to simulate video traffic are assigned to the *video* class
- All IP control protocol traffic with the DSCP values 48 and 56 are assigned to the *network-internetwork-control* class
- The rest of the IP traffic is assigned to the default class
- All CPU-generated non-IP traffic is statically mapped to a fixed queue on the egress port
- All CFM traffic is queued to the default class because there is no class based on CoS

```
Switch(config)# cpu traffic qos dscp dscp
```

Class

```
Switch(config)# class-map match-any video
Switch(config-cmap)# match ip dscp af41 af42 af43
Switch(config-cmap)# exit

Switch(config)# class-map match-any voice
Switch(config-cmap)# match ip dscp ef
Switch(config-cmap)# exit

Switch(config)# class-map match-any network-internetwork-control
Switch(config-cmap)# match ip dscp 48 56
Switch(config-cmap)# exit
```

Policy

```
Switch(config)# policy-map output-policy
Switch(config-pmap)# class voice
Switch(config-pmap-c)# priority
Switch(config-pmap-c)# police cir 10000000
Switch(config-pmap-c)# exit
Switch(config-pmap)# class video
Switch(config-pmap-c)# bandwidth percent 40
Switch(config-pmap-c)# exit
Switch(config-pmap)# class network-internetwork-control
Switch(config-pmap-c)# bandwidth percent 10
Switch(config-pmap-c)# exit
Switch(config-pmap)# class class-default
Switch(config-pmap-c)# bandwidth percent 30
Switch(config-pmap-c)# exit
```

Interface

```
Switch(config)# interface fastethernet0/1
Switch(config-if)# service-policy output output-policy
Switch(config-pmap-c)# exit
```

Example 2

This example shows how to mark the CoS of CPU-generated IP traffic (including IP-SLA and TWAMP) based on the DSCP value in the packet and to configure egress queuing based on the CoS value.

- All CPU-generated IP traffic queues on the egress port, based on the IP DSCP value and the configured output policy map called *output-policy*.
- All IP SLA or TWAMP probes with the DSCP value *ef* to simulate voice traffic are assigned to the *voice* class
- All IP SLA or TWAMP probes with the DSCP values *af41*, *af42* and *af43* to simulate video traffic are assigned to the *video* class
- All IP control protocol traffic with the DSCP values 48 and 56 are assigned to the *network-internetwork-control* class
- The rest of the IP traffic is assigned to the default class
- All CPU-generated non-IP traffic with CoS 5 is assigned to the *voice* class
- All CPU-generated non-IP traffic with CoS 3 is assigned to the *video* class
- All CPU-generated non-IP traffic with CoS 6 and 7 is assigned to the *network-internetwork-control* class
- All CFM traffic with CoS 5 is assigned to the *voice* class
- All CFM traffic with CoS 3 is assigned to the *video* class
- All CFM traffic with CoS 6 and 7 is assigned to the *network-internetwork-control* class

Table Map

```
Switch(config)# table-map dscp-to-cos
Switch(config-tablemap)# map from 46 to 5
Switch(config-tablemap)# map from 48 to 6
Switch(config-tablemap)# map from 56 to 7
Switch(config-tablemap)# map from af41 to 3
Switch(config-tablemap)# map from af42 to 3
Switch(config-tablemap)# map from af43 to 3
Switch(config-tablemap)# default 0
Switch(config-tablemap)# end
```

CPU QoS

```
Switch(config)# cpu traffic qos cos dscp table-map dscp-to-cos
Switch(config)# cpu traffic qos cos cos
```

Class

```
Switch(config)# class-map match-any video
Switch(config-cmap)# match cos 3
Switch(config-cmap)# exit

Switch(config)# class-map match-any voice
Switch(config-cmap)# match cos 5
Switch(config-cmap)# exit

Switch(config)# class-map match-any network-internetwork-control
Switch(config-cmap)# match cos 6 7
Switch(config-cmap)# exit
```

Policy

```
Switch(config)# policy-map output-policy
Switch(config-pmap)# class voice
Switch(config-pmap-c)# priority
Switch(config-pmap-c)# police cir 10000000
Switch(config-pmap-c)# exit
Switch(config-pmap)# class video
Switch(config-pmap-c)# bandwidth percent 40
Switch(config-pmap-c)# exit
Switch(config-pmap)# class network-internetwork-control
Switch(config-pmap-c)# bandwidth percent 10
Switch(config-pmap-c)# exit
Switch(config-pmap)# class class-default
Switch(config-pmap-c)# bandwidth percent 30
Switch(config-pmap-c)# exit
```

Interface

```
Switch(config)# interface fastethernet0/1
Switch(config-if)# service-policy output output-policy
Switch(config-pmap-c)# exit
```

Example 3

This example shows how to:

- Mark the DSCP value of CPU-generated IP traffic (including IP-SLA and TWAMP) based on the DSCP value in the packet
- Mark the CoS of CPU-generated IP traffic (including IP-SLA and TWAMP) based on the DSCP value in the packet
- Mark the CoS of CPU-generated non-IP traffic based on the CoS value in the packet.
- Mark all CPU-generated traffic with the QoS group
- Configure egress queuing based on the QoS group

The example has these results:

- All CPU-generated IP traffic with DSCP values 46, 48, and 56 retains the existing markings
- For all other CPU-generated IP packets, the DSCP value is reset to 0
- All CPU-generated IP traffic with DSCP values 46, 48, and 56 is mapped to the corresponding CoS values of 5, 6, and 7 respectively
- For all other CPU-generated IP packets, the CoS value resets to 0

- All CPU-generated non-IP traffic with the CoS values of 5, 6, and 7 retain the existing markings.
- For all other CPU-generated non-IP packets, the CoS value resets to 0
- All CPU-generated traffic goes through a single class called *cpu-traffic*. The *user-voice* classes *user-voice* and *user-video* are reserved for user traffic. As a result, CPU traffic and user traffic are separated into different queues on the egress port.

Table Map

```
Switch(config)# table-map dscp-to-cos
Switch(config-tablemap)# map from 46 to 5
Switch(config-tablemap)# map from 48 to 6
Switch(config-tablemap)# map from 56 to 7
Switch(config-tablemap)# default 0
Switch(config-tablemap)# end

Switch(config)# table-map dscp-to-dscp
Switch(config-tablemap)# map from 46 to 46
Switch(config-tablemap)# map from 48 to 48
Switch(config-tablemap)# map from 56 to 56
Switch(config-tablemap)# default 0
Switch(config-tablemap)# end

Switch(config)# table-map cos-to-cos
Switch(config-tablemap)# map from 5 to 5
Switch(config-tablemap)# map from 6 to 6
Switch(config-tablemap)# map from 7 to 7
Switch(config-tablemap)# default 0
Switch(config-tablemap)# end
```

CPU QoS

```
Switch(config)# cpu traffic qos dscp dscp table-map dscp-to-dscp
Switch(config)# cpu traffic qos cos dscp table dscp-to-cos
Switch(config)# cpu traffic qos cos cos table cos-to-cos
Switch(config)# cpu traffic qos qos-group 50
```

Class

```
Switch(config)# class-map match-any cpu-traffic
Switch(config-cmap)# match qos-group 50
Switch(config-cmap)# exit

Switch(config)# class-map match-any user-video
Switch(config-cmap)# match cos 3
Switch(config-cmap)# exit

Switch(config)# class-map match-any user-voice
Switch(config-cmap)# match cos 5
Switch(config-cmap)# exit
```

```
Policy
```

```
Switch(config)# policy-map output-policy
Switch(config-pmap)# class user-voice
Switch(config-pmap-c)# priority
Switch(config-pmap-c)# police cir 10000000
Switch(config-pmap-c)# exit
Switch(config-pmap)# class user-video
Switch(config-pmap-c)# bandwidth percent 40
Switch(config-pmap-c)# exit
Switch(config-pmap)# class cpu-traffic
Switch(config-pmap-c)# bandwidth percent 10
Switch(config-pmap-c)# exit
Switch(config-pmap)# class class-default
Switch(config-pmap-c)# bandwidth percent 30
Switch(config-pmap-c)# exit

Interface

Switch(config)# interface fastethernet0/1
Switch(config-if)# service-policy output output-policy
Switch(config-pmap-c)# exit
```

Displaying QoS Information

To display QoS information, use one or more of the privileged EXEC commands in Table 14-2. For explanations about available keywords, see the command reference for this release.

Table 14-2 Commands for Displaying QoS Information

Command	Description
show class-map [*class-map-name*]	Display QoS class-map information for all class maps or the specified class map.
show policer aggregate [*aggregate-policer-name*]	Display information about all aggregate policers or the specified aggregate policer.
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 cpu traffic qos	Display the QoS marking values for CPU-generated traffic.
show running-config	Display the configured class maps, policy maps, table maps, and aggregate policers.
show table-map [*table-map-name*]	Display information for all configured table maps or the specified table map.
show class-map [*class-map-name*]	Display QoS class-map information for all class maps or the specified class map.

To test full-path QoS in both directions on an interface, you can configure Ethernet terminal loopback by entering the **ethernet loopback facility** interface configuration command. In terminal loopback mode, the port appears to be up but the link is actually down and no packets are sent out. Configuration changes on the port immediately affect the traffic being looped back. For information about Ethernet terminal loopback, see the “Enabling Ethernet Loopback” section in Chapter 45, “Configuring Ethernet OAM, CFM, and E-LMI” in the *Cisco CGS 2520 Switch Software Configuration Guide*:

http://www.cisco.com/en/US/docs/switches/connectedgrid/cgs2520/software/release/12_2_53_ex/configuration/guide/swoam.html#wp1137522

QoS Statistics

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

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

This output also includes byte-level statistics for conform, exceed, and violate classes.

Another way to view input QoS statistics is in the output of the **show platform qos statistics interface** [*interface-id*] privileged EXEC command. The per-port frame statistics are sorted by the DSCP and CoS values of the incoming frames on the port. These statistics do not provide any information about the MQC input policy map configured on the interface.

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

Configuration Examples for Policy Maps

This section includes configuration examples for configuring QoS policies on the switch module, including configuration limitations and restrictions. The sections are broken into different configurations actions that a customer might do. Each section provides the exact sequence of steps that you must follow for successful configuration or modification.

- QoS Configuration for Customer A, page 14-87
- QoS Configuration for Customer B, page 14-89
- Modifying Output Policies and Adding or Deleting Classification Criteria, page 14-90
- Modifying Output Policies and Changing Queuing or Scheduling Parameters, page 14-91
- Modifying Output Policies and Adding or Deleting Configured Actions, page 14-92
- Modifying Output Policies and Adding or Deleting a Class, page 14-93

QoS Configuration for Customer A

This section provides examples of the initial configuration and activation of QoS policies for a customer switch module. Input and output QoS service policies are configured based on the requirements and attached to relevant ports.

In the initial configuration for Customer A, Fast Ethernet ports 1 through 24 are user network interfaces (UNIs) and are disabled by default. Gigabit Ethernet ports 1 and 2 are network node interfaces (NNIs) and are enabled by default.

This is the overall sequence for initial configuration:

- Configure classes and policies.
- Shut down all active ports.

- Attach policies to ports to be activated.
- Take the ports out of the shut-down state.
- Leave unused ports shut down.

Note these restrictions for configuring output policies:

- You can define up to three classes in the output policy map.
- The defined classes must be the same as other output policy maps.
- The number of defined classes in each output policy map must be same.
- You must assign an action to each class; that is, there can be no empty class.
- Each class configuration must be based on the classification/marking done in the input policy-map.

This example configures classes for input service policies and defines three classes of service: gold, silver, and bronze. Because a **match-all** classification (the default) can have only single classification criterion, the **match-any** classification is used so that you can add classification criteria in the future:

```
Switch# config terminal
Switch(config)# class-map match-any gold-in
Switch(config-cmap)# match ip dscp af11
Switch(config-cmap)# exit
Switch(config)# class-map match-any silver-in
Switch(config-cmap)# match ip dscp af21
Switch(config-cmap)# exit
Switch(config)# class-map match-any bronze-in
Switch(config-cmap)# match ip dscp af31
Switch(config-cmap)# exit
```

This example shows how to configure an input policy map that marks the gold class and polices the silver class to 50 Mb/s and the bronze class to 20 Mb/s.

```
Switch(config)# policy-map input-all
Switch(config-pmap)# class gold-in
Switch(config-pmap-c)# set ip dscp af43
Switch(config-pmap-c)# exit
Switch(config-pmap)# class silver-in
Switch(config-pmap-c)# police 50000000
Switch(config-pmap)# class bronze-in
Switch(config-pmap-c)# police 20000000
Switch(config-pmap-c)# exit
```

This example configures classes for output service policies with three classes of service: gold, silver, and bronze. The gold class is configured to match the marked value in the input service policy. Because a **match-all** classification (the default) can have only single classification criterion, the **match-any** classification is used so that you can add classification criteria in the future.

```
Switch# config terminal
Switch(config)# class-map match-any gold-out
Switch(config-cmap)# match ip dscp af43
Switch(config-cmap)# exit
Switch(config)# class-map match-any silver-out
Switch(config-cmap)# match ip dscp af21
Switch(config-cmap)# exit
Switch(config)# class-map match-any bronze-out
Switch(config-cmap)# match ip dscp af31
Switch(config-cmap)# exit
```

This example configures one output service policy to be applied to both Gigabit Ethernet NNIs, providing priority with rate-limiting to the gold class, class-based shaping for the silver class, and a minimum bandwidth guarantee of 10 percent to the bronze class.

```
Switch(config)# policy-map output-g1-2
Switch(config-pmap)# class gold-out
Switch(config-pmap-c)# priority
Switch(config-pmap-c)# police 50000000
Switch(config-pmap-c)# exit
Switch(config-pmap)# class silver-out
Switch(config-pmap-c)# shape average 200000
Switch(config-pmap-c)# exit
Switch(config-pmap)# class bronze-out
Switch(config-pmap-c)# bandwidth percent 10
Switch(config-pmap-c)# exit
```

This example configures a second output service policy to be applied to Fast Ethernet UNIs 1 to 8, providing strict priority to the gold class and distributing the remaining bandwidth in the desired proportions over the remaining classes.

```
Switch(config)# policy-map output1-8
Switch(config-pmap)# class gold-out
Switch(config-pmap-c)# priority
Switch(config-pmap-c)# exit
Switch(config-pmap)# class silver-out
Switch(config-pmap-c)# bandwidth remaining percent 50
Switch(config-pmap-c)# exit
Switch(config-pmap)# class bronze-out
Switch(config-pmap-c)# bandwidth remaining percent 20
Switch(config-pmap-c)# exit
```

This example attaches the input and output service policies to the Gigabit Ethernet ports and activates them.

```
Switch(config)# interface range gigabitethernet0/1-2
Switch(config-if-range)# service-policy input input-all
Switch(config-if-range)# service-policy output output-g1-2
Switch(config-if-range)# no shutdown
Switch(config-if-range)# exit
```

This example attaches the input and output service policies to Fast Ethernet ports 1 to 8 and activates them.

```
Switch(config)# interface range fastethernet0/1 - 8
Switch(config-if-range)# service-policy input input-all
Switch(config-if-range)# service-policy output output1-8
Switch(config-if-range)# no shutdown
Switch(config-if-range)# exit
```

QoS Configuration for Customer B

This section provides examples for configuring and activating QoS policies on the switch module for a new set of customers without affecting the current customers. Input and output QoS service policies are configured based on the requirements and attached to relevant ports. The example uses an existing input policy-map and configures a new output policy map for the new customers.

In the initial configuration for Customer B, Fast Ethernet ports 1 through 8 are UNIs and are active. Fast Ethernet ports 9 through 24 are UNIs and are shut down. Gigabit Ethernet ports 1 and 2 are NNIs and are enabled by default.

This is the overall sequence of initial configuration:

- Define any new required output policies
- Attach input and output policies to ports to be activated
- Take the ports out of the shut-down state

Note these restrictions when configuring output policies:

- You can define up to three classes in the output policy map
- The defined classes must be the same as other output policy maps
- The number of defined classes in each output policy map must be same
- You must assign an action to each class; that is, there can be no empty class
- Each class configuration must be based on the classification/marking done in the input policy-map.

This example configures a third output service policy to be attached to Fast Ethernet UNIs 9 through 12, providing a minimum guaranteed bandwidth of 50 Mb/s to the gold class, 20 Mb/s to the silver class, and 10 Mb/s to the bronze class:

```
Switch(config)# policy-map output9-12
Switch(config-pmap)# class gold-out
Switch(config-pmap-c)# bandwidth 50000
Switch(config-pmap-c)# exit
Switch(config-pmap)# class silver-out
Switch(config-pmap-c)# bandwidth 20000
Switch(config-pmap-c)# exit
Switch(config-pmap)# class bronze-out
Switch(config-pmap-c)# bandwidth 10000
Switch(config-pmap-c)# exit
```

This example attaches the output policy for Fast Ethernet ports 9 through 12 and activates the ports:

```
Switch# config terminal
Switch(config)# interface range fastethernet0/9-12
Switch(config-if-range)# service-policy input input-all
Switch(config-if-range)# service-policy output output9-12
Switch(config-if-range)# no shutdown
Switch(config-if-range)# exit
```

Modifying Output Policies and Adding or Deleting Classification Criteria

This section provides examples of updating an existing set of output policy maps to add or delete classification criteria. The modification might be required due to a change in the service provisioning requirements or a change in the input service policy map. You can make the change without shutting down any port.

In the initial configuration, Fast Ethernet ports 1 through 12 are UNIs and are active. Fast Ethernet ports 13 through 24 are UNIs and are shut down. Gigabit Ethernet ports 1 and 2 are NNIs and are enabled by default.

This is the overall sequence of initial configuration:

- Change the configured class map for an input service policy

- Change the configured class map for an output service policy

This example modifies classes for an input service policy by adding classification criteria to the silver-in class to also match dscp cs5. This is required for the output policy-map to match to dscp cs5:

```
Switch(config)# class-map match-any silver-in
Switch(config-cmap)# match ip dscp af21
Switch(config-cmap)# match ip dscp cs5
Switch(config-cmap)# exit
```

This example modifies classes for an output service policy, adding classification criteria to the silver-out class to also match dscp cs5. This adds dscp cs5 to the silver-out class on all configured and attached output service policies. The dscp cs5 flow now receives the same queuing and scheduling treatment as the silver-out class.

```
Switch# config terminal
Switch(config)# class-map match-any silver-out
Switch(config-cmap)# match ip dscp af21
Switch(config-cmap)# match ip dscp cs5
Switch(config-cmap)# exit
```

You should use the same procedure when deleting a match statement associated with a configured class.

Modifying Output Policies and Changing Queuing or Scheduling Parameters

This section provides examples of updating an existing set of output policy maps to modify the parameters of the configured queuing and scheduling actions. The modification in the output policy map might be required due to a change in the service provisioning requirements. You can make the change without shutting down any port.

In the initial configuration, Fast Ethernet ports 1 through 12 are UNIs and are active. Fast Ethernet ports 13 through 24 are UNIs and are shut down. Gigabit Ethernet ports 1 and 2 are NNIs and are enabled by default.

The requirement is to change the action parameters.

Output Policy Restrictions

Note these restrictions when configuring output policies:

- You can define up to three classes in the output policy map
- The defined classes must be the same as other output policy maps
- The number of defined classes in each output policy map must be same
- You must assign an action to each class; that is, there can be no empty class
- Each class configuration must be based on the classification or marking done in the input policy-map

This example modifies the third output service policy servicing Fast Ethernet UNIs 8 through 12 by providing minimum guaranteed bandwidth of 40 Mb/s to the gold class (changed from 50 Mb/s), 30 Mb/s to the silver class (changed from 20 Mb/s), and 20 Mb/s to the bronze class (changed from 10 Mbps).

```
Switch(config)# policy-map output9-12
Switch(config-pmap)# class gold-out
Switch(config-pmap-c)# bandwidth 40000
Switch(config-pmap-c)# exit
Switch(config-pmap)# class silver-out
Switch(config-pmap-c)# bandwidth 30000
Switch(config-pmap-c)# exit
Switch(config-pmap)# class bronze-out
Switch(config-pmap-c)# bandwidth 20000
Switch(config-pmap-c)# exit
```

Modifying Output Policies and Adding or Deleting Configured Actions

This section provides examples of updating an existing set of output policy maps to add or delete queuing and scheduling actions. The modification in the output policy map might be required due to a change in the service provisioning requirements. You can make the change without shutting down ports that are not configured with the output policy map to be modified. But you must shut down the ports that are configured with that output policy map. Customers not using this output policy map are not affected.

In the initial configuration, Fast Ethernet ports 1 through 12 are UNIs and are active. Fast Ethernet ports 13 through 24 are UNIs and are shut down. Gigabit Ethernet ports 1 and 2 are NNIs and are enabled by default.

This is the overall sequence of configuration:

- Shut down all active ports carrying the policy to be modified
- Detach the output policy from all ports to which it is attached
- Make modifications to the output policy
- Reattach the output policy to the appropriate ports
- Take the ports out of the shutdown state

Note these restrictions for configuring output policies:

- You can define up to three classes in the output policy map
- The defined classes must be the same as other output policy maps
- The number of defined classes in each output policy map must be same
- You must assign an action to each class; that is, there can be no empty class
- Each class configuration must be based on the classification/marking done in the input policy-map

These steps shut down all ports carrying the output policy, in this case only the Gigabit Ethernet ports

```
Switch(config)# interface range gigabitEthernet0/1-2
Switch(config-if-range)# shutdown
Switch(config-if-range)# exit
```

.These steps detach the output policy to be modified, in this case the one configured on the Gigabit Ethernet ports:

```
Switch(config)# interface range gigabitEthernet0/1-2
Switch(config-if-range)# no service-policy output output-g1-2
Switch(config-if-range)# exit
```

These steps modify the output service policy servicing the Gigabit Ethernet NNIs. Instead of providing a minimum bandwidth guarantee of 10 percent to the bronze class, the policy is modified to provide class-based shaping to 100000 bps.

```
Switch(config)# policy-map output-g1-2
Switch(config-pmap)# class bronze-out
Switch(config-pmap-c)# no bandwidth percent 10
Switch(config-pmap-c)# shape average 100000
Switch(config-pmap-c)# exit
```

These steps reattach the output policy to the Gigabit Ethernet ports:

```
Switch(config)# interface range gigabitethernet0/1-2
Switch(config-if-range)# service-policy output output9-12
Switch(config-if-range)# exit
```

These steps activate all Gigabit Ethernet ports:

```
Switch(config)# interface range gigabitethernet0/1-2
Switch(config-if-range)# no shutdown
Switch(config-if-range)# exit
```

Modifying Output Policies and Adding or Deleting a Class

This section provides examples of updating an existing set of output policy maps to add or delete entire classes. The modification in the output policy map might be required due to a change in the service provisioning requirements or a change in the input service policy. To make this change, you must shut down all active ports on the switch module. For this kind of update to any output policy map, all customers could potentially be affected. To avoid this, we recommend that you consider possible future upgrades when you configure classes in output service policies.

In the initial configuration, Fast Ethernet ports 1 through 12 are UNIs and are active. Fast Ethernet ports 13 through 24 are UNIs and are shut down. Gigabit Ethernet ports 1 and 2 are NNIs and are enabled by default.

This is the overall sequence of configuration:

- Shut down all active ports.
- Detach the output policies from all Fast Ethernet and Gigabit Ethernet ports.
- Delete the class.
- Reattach the output policies to the Fast Ethernet and Gigabit Ethernet ports.
- Take the Fast Ethernet and Gigabit Ethernet ports out of the shutdown state.

These steps shut down all active and applicable Fast Ethernet and Gigabit Ethernet ports:

```
Switch(config)# interface range gigabitethernet0/1-2, fastethernet0/1-12
Switch(config-if-range)# shutdown
Switch(config-if-range)# exit
```

These steps detach all output policies from the affected Fast Ethernet and Gigabit Ethernet ports:

```
Switch(config)# interface range fastethernet0/1-8
Switch(config-if-range)# no service-policy output output1-8
Switch(config-if-range)# exit

Switch(config)# interface range fastethernet0/9-12
Switch(config-if-range)# no service-policy output output9-12
Switch(config-if-range)# exit

Switch(config)# interface range gigabitethernet0/1/0-2
Switch(config-if-range)# no service-policy output output-g1-2
Switch(config-if-range)# exit
```

These steps delete a class from all output policy maps and input policy maps; the input policy can be left attached or can be detached:

```
Switch(config)# policy-map output1-8
Switch(config-pmap)# no class bronze-out
Switch(config-pmap-c)# exit
Switch(config)# policy-map output9-12
Switch(config-pmap)# no class bronze-out
Switch(config-pmap-c)# exit
Switch(config)# policy-map output-g1-2
Switch(config-pmap)# no class bronze-out
Switch(config-pmap-c)# exit
Switch(config)# policy-map input-all
Switch(config-pmap)# no class bronze-in
Switch(config-pmap-c)# exit
```

These steps reattach all policies to the Fast Ethernet and Gigabit Ethernet ports:

```
Switch(config)# interface range fastethernet0/1-8
Switch(config-if-range)# service-policy output output1-8
Switch(config-if-range)# exit

Switch(config)# interface range fastethernet0/9-12
Switch(config-if-range)# service-policy output output9-12
Switch(config-if-range)# exit

Switch(config)# interface range gigabitethernet0/1/0-2
Switch(config-if-range)# service-policy output output9-12
Switch(config-if-range)# exit
```

These steps activate all applicable Fast Ethernet and Gigabit Ethernet ports:

```
Switch(config)# interface range gigabitethernet0/1-2, fastethernet0/1-12
Switch(config-if-range)# no shutdown
Switch(config-if-range)# exit
```

You should use the same procedure when adding a class to an attached output service policy.

Note Problems can occur if you do not follow the previous sequence.

When a policy map is attached to an interface, all traffic that does not explicitly match the configured class maps within the policy map should go through the default queue (class **class-default**). However, in some cases, traffic that does not explicitly match the output policy-map classes could go through more than one queue. This queuing problem can occur when you do not follow the previous procedure and do not attach an output policy to all active ports.

For example, consider this case where only two ports are configured with an output policy and we want to delete a class in the output policy.

Shut down two ports:

```
Switch(config)# interface range fastethernet0/1-2
Switch(config-if-range)# shutdown
Switch(config-if-range)# exit
```

Detach the output policy from both ports:

```
Switch(config)# interface range fastEthernet0/1-2
Switch(config-if)# no service-policy output output1-2
Switch(config-if)# exit
```

Delete a class in the output policy:

```
Switch(config)# policy-map output1-2
Switch(config-pmap)# no class bronze-out
Switch(config-pmap-c)# exit
```

Attach the output policy to only one port and not to the other:

```
Switch(config)# interface FastEthernet0/1
Switch(config-if)# service-policy output output1-2
Switch(config-if)# exit
```

Enable both ports:

```
Switch(config)# interface range fastethernet0/1-2
Switch(config-if-range)# no shutdown
Switch(config-if-range)# exit
```

At this point, when traffic leaves Fast Ethernet port 2, instead of going through a single default-queue, it goes through the same number of queues as there are classes defined in the output policy-map attached to Fast Ethernet port 1. In this case, it would be three queues. In some cases, packets for a flow out of Fast Ethernet port 2 might be reordered if a flow splits across more than one queue. You can avoid this problem by leaving ports in a shut-down state until you attach an output policy.

Implementing High-Priority Traffic to the Host Router

This section describes QoS features for implementing high-priority (low-latency) traffic via the internal data path between the CGR 2010 ESM and the host CGR 2010 router. The internal data path is called *PortChannel48*. For details, see Chapter 9, "EtherChannel Configuration Between the Switch Module and the Host Router."

Note In this section, *ingress* refers to the direction of the data flow from the CGR 2010 ESM to the host router and *egress* refers to the direction of the data flow from the host router to the switch module.

Tip QoS configuration for the switch module is concerned only with the ingress traffic from the switch module to the host router.

You can configure QoS on the switch module and the CGR 2010 router using Modular QoS CLI (MQC), with some limitations on the switch module. The following subsections describe the major components in this internal data path and their QoS functions.

Ingress CoS to GDF Queue Mapping

In a substation network, some data frames (for example, GOOSE (Generic Object Oriented Substation Events) messages) require low-latency in the data path. The switch module provides high priority queuing to support low-latency data. This section describes how low-latency data frames from the switch module are mapped to the Generic Data Frame (GDF) queues.

Unlike the switch module, which uses a hardware ASIC to perform Layer 2 to Layer 3 switching, the CGR 2010 router's data path is implemented in software and can potentially have greater jitter and delay in its data path. If all data packets from the switch module are sent to a single GDF queue in the CGR 2010 router, low-latency data could be blocked by the other low priority data frames in the same queue.

To prevent low-latency data frames from being blocked by other lower priority data frames on the ingress data path to the host CGR 2010 router, the switch module sends high priority data frames and the other lower priority data frames to two separate GDF queues:

- The GDF queue for low-latency (high-priority) data is called the *priority queue* (GDF queue 0).
- The GDF queue for other data is called the *low-priority queue* (GDF queue 1).

Network administrators can use MQC to classify and/or mark a data frame based on the information on the Layer 2 and Layer 3 headers. *The mapping of data frames to GDF queues is solely based on the CoS field in the 802.1Q tag.* Data frames that require low latency in the internal data path from the switch module to the host router should contain an 802.1Q tag and be marked with CoS values that are configured with priority queuing (with or without policing).

All other data frames, which either do not contain an 802.1Q tag or their CoS values are not configured with priority queuing, are mapped to the low-priority queue. For details and example configurations, see "Mapping CoS to the High-Priority GDF Queue" section on page 14-97.

Adjusting for Differences in Ingress Traffic Bandwidth

The CGR 2010 router processes incoming and outgoing packets in software. Depending on the performance of the router's CPU, the effective bandwidth of the internal data path on the CGR 2010 router's side could be less than the bandwidth provided by the internal EtherChannel on the switch module.

If the effective bandwidth on the CGR 2010 router is less than the bandwidth of the internal EtherChannel, you should use the **MQC** command to configure port shaping on the internal EtherChannel on the switch module so that the effective bandwidth on the two sides are roughly equivalent. For more information, see the "Congestion Management and Scheduling" section on page 14-26.

Configuring CPU-Generated Data

The switch module can send control data to the CGR 2010 host router through the backplane PortChannel48. You can configure QoS for the CPU-generated data frames with MQC. If you would like to send the CPU-generated data frames to the GDF priority queue, you have to configure the CPU so that all CPU-generated data would be marked with a non-zero CoS value.

To do so, use the **cpu traffic qos cos** command in global configuration mode to configure quality of service (QoS) marking based on class of service (CoS) for control plane traffic. To return to the default value, use the **no** form of this command.

```
cpu traffic qos cos {cos_value }
```

For example:

```
Switch(config)# cpu traffic qos cos 5
```

Syntax	Description
cos-value	Specify a CoS value. The range is from 0 to 7. If no CoS value is configured, the protocol-specific default value for each packet is applied.
cos	Configure the CoS value based on the CoS value in the packet, using a table-map.

Usage Guidelines

Configure any desired table-maps before configuring marking or queuing of CPU traffic.

This feature must be configured globally for a switch module; it cannot be configured per-port or per-protocol.

Enter each cpu traffic qos marking action on a separate line.

The **cpu traffic qos cos** global configuration command configures CoS marking for CPU-generated traffic by using either a specific CoS value or a table map, but not both. A new configuration overwrites the existing configuration.

When the **cpu traffic qos cos** global configuration command is configured with table maps, you can configure two **map from** values at a time—*CoS* and either *DSCP* or *precedence*.

Mapping CoS to the High-Priority GDF Queue

As mentioned earlier in this section, GDF queue mapping is solely based on the CoS value in the 802.1Q tag. Untagged data frames are mapped to the GDF low-priority queue. This imposes some limitations on how users configure QoS on the internal PortChannel interface in the CGR 2010 ESM.

A *native VLAN* is the untagged VLAN on an 802.1Q trunked switch module port. The native VLAN and management VLAN could be the same, but it is better security practice that they are not. If a switch module receives untagged frames on a trunk port, they are assumed to be part of the VLAN that is designated on the switch module port as the native VLAN. Frames egressing a switch module port on the native VLAN are not tagged.

By default, the data frames belonging to a native VLAN are untagged. If users would like to configure priority queuing for native VLAN, they should use the **vlan dot1q tag native** CLI command to enable 802.1Q tagging for native VLAN.

Note **vlan dot1q tag native** is a global configuration command and applies to all trunk ports in the switch module.

Please note the following:

- Users should not configure priority queuing for data frames marked with CoS 0 because it is the default CoS value for data frames whose CoS value are not marked.
- Low-latency traffic destined to the GDF priority queue should be marked with a non-zero CoS value and configured with priority queuing using MQC (see the "Modular QoS CLI" section on page 14-3).
- Chosen CoS values for priority queuing applies to the data frame for all VLANs

Example Configuration for CoS to GDF Queue Mapping

This section provides an example configuration to map CoS 5 to GDF queue 0 (the high-priority queue) for the internal PortChannel 48 on the CGR 2010 ESM copper model (GRWIC-D-2S-8PC).

	Step	Command
Step 1	Enters global configuration mode.	**configure terminal**
Step 2	Creates a class map, and enters class-map configuration mode. For details on using the **class-map** command, see the "Using Class Maps to Define a Traffic Class" section on page 14-42. • In this example, we 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. • In this example, *class0* specifies the name of the class map.	**class-map match-all class0**
Step 3	Defines the match criterion to classify traffic. Only one match type per class map is supported, and only one ACL per class map is supported. • The **cos 5** parameter specifies a value of **5** for the CoS field to match against incoming packets.	**match cos 5**
Step 4	Creates a policy map by entering the policy map name (for example, *policy0*), and enters policy-map configuration mode.	**policy-map policy0**
Step 5	Enters the name of the priority class (created by using the **class-map** global configuration command shown in Step 2), and enters policy-map class configuration mode for the priority class.	**class class0**
Step 6	Sets the strict scheduling priority for this class.	**priority**
Step 7	Specifies the interface to configure, and enters interface configuration mode.	**interface FastEthernet0/9**
Step 8	Configures the switch's PortChannel as a Layer 2 VLAN trunk, so that the specified port can transport the frames for multiple VLANs.	**switchport mode trunk**
Step 9	The **service-policy output policy0** interface configuration command attaches all the characteristics of the traffic policy named **policy0** to the specified interface. All packets leaving the specified interface are evaluated according to the criteria specified in the traffic policy named **policy0**.	**service-policy output policy0**

	Step	Command
Step 10	Repeat Steps 7 through 9 for each interface in the PortChannel48 (a total of eight interfaces).	
Step 11	Return to privileged EXEC mode.	**end**
Step 12	(Optional) Save your entries in the configuration file.	**copy running-config startup-config**