



CHAPTER 9

Configuring QoS on the Optical Services Modules

This chapter describes how to configure Quality of Service (QoS) on the Optical Services Modules (OSMs).

This chapter consists of these sections:

- [Understanding QoS on the OSMs, page 9-1](#)
- [Configuring QoS on the OSMs, page 9-2](#)
- [Unsupported Frame Relay-Specific QoS Features, page 9-24](#)
- [Cisco IPv6 QoS on the OSMs, page 9-24](#)

Understanding QoS on the OSMs

QoS for the OSMs is distributed between the OSMs and the Policy Feature Card. This chapter discusses the Layer 3 QoS implementations on the OSM WAN ports.

For the OSM WAN ports, the OSMs support the following Layer 3 QoS implementations:

- Class-based traffic shaping
- Class-based weighted fair queuing (CBWFQ)
- Low latency queuing (LLQ)
- Weighted Random Early Detection (WRED)
- Hierarchical Traffic Shaping

You configure OSM WAN port QoS using a subset of the Modular QoS CLI (MQC). For information on MQC, refer to the *Modular Quality of Service Command-Line Interface Overview* at:

<http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120limit/120xe/120xe5/mqc/mcli.htm>.



Note

Though the OSM-2OC12-POS-SI+ card contains 2 POS ports and 4 GigE ports, the GigE ports do not support mqc queuing or shaping.



Note

Using the **fair-queue** and **random-detect** commands on the main interface is not supported with the OSMs.

For OSM WAN port Layer 3 QoS configuration information and examples, see the “Configuring QoS on the OSMs” section on page 9-2 and the [Configuring MPLS QoS, page 11-13](#).

Additional QoS Features and Resources

- For information about configuring AToM QoS on the OSM WAN ports, see the “[How to Configure QoS with AToM](#)” section on page 11-78.
- For information about configuring the Catalyst Layer 2 QoS implementation for 1p1q4t ingress queues and 1p2q2t egress queues on the OSM Gigabit Ethernet LAN ports, refer to the links below for Policy Feature Card QoS on the Cisco 7600 series router and the Catalyst 6000 series switch.
- For information about configuring the Policy Feature Card for Layer 2 and Layer 3 policing and marking of traffic for the OSM WAN ports and the OSM Gigabit Ethernet LAN ports, refer to the links below for Policy Feature Card QoS on the Cisco 7600 series router and the Catalyst 6000 series switch.



Note The Policy Feature Cards 3BXL and PFC3B add support for MPLS policing and marking.

- For information about configuring Policy Feature Card QoS on the Cisco 7600 series router, see:
 - *Cisco 7600 Series Cisco IOS Software Configuration Guide, 12.2SX*
<http://www.cisco.com/en/US/docs/routers/7600/ios/12.2SXF/configuration/guide/swcg.html>
- For information about configuring Policy Feature Card QoS on the Catalyst 6000 series switch running Cisco IOS software on the supervisor engine and on the MSFC, see:
 - *Catalyst 6500 Series Cisco IOS Software Configuration Guide, 12.2SX*
<http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2SX/configuration/guide/book.html>
 - *Catalyst 6500 Series Cisco IOS Command Reference, 12.2SX* at this URL:
http://www.cisco.com/en/US/docs/ios/mcl/122sxmcl/12_2sx_mcl_book.html
- For general information on how to configure Cisco IOS QoS, see:
 - *Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.2*
http://www.cisco.com/en/US/docs/ios/12_2/qos/configuration/guide/fqos_c.html
 - *Cisco IOS Quality of Service Solutions Command Reference, Release 12.2*
http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fqos_r/index.htm

Configuring QoS on the OSMs

These sections describe configuring QoS on the OSMs:

- [Enabling QoS Globally, page 9-3](#)
- [Configuring Classification, page 9-3](#)
- [Configuring Class-Based Traffic Shaping, page 9-4](#)
- [Configuring Class-Based Weighted Fair Queuing, page 9-7](#)
- [Configuring Low Latency Queuing, page 9-12](#)
- [Configuring Weighted Random Early Detection, page 9-15](#)

- [Configuring Hierarchical Traffic Shaping, page 9-17](#)
- [Configuring Queue Limit, page 9-19](#)
- [Configuring QoS: Match VLAN, page 9-20](#)
- [Distribution of Remaining Bandwidth, page 9-22](#)

For information on shaping features supported with Destination Sensitive Services (DSS), see [Chapter 10, “Configuring Destination Sensitive Services on the Optical Services Modules.”](#)

Enabling QoS Globally

Before you can configure QoS on the OSMs, you must enable the QoS functionality globally. To enable QoS globally, perform this task:

	Command	Purpose
Step 1	Router(config)# mls qos	Enables QoS on the switch. Use the no mls qos command to globally disable QoS.
Step 2	Router(config)# end	Exits configuration mode.
Step 3	Router# show mls qos	Verifies the configuration.

This example shows how to enable QoS globally:

```
Router(config)# mls qos
Router(config)# end
Router#
```

This example shows how to verify the configuration:

```
Router# show mls qos
  QoS is enabled globally
  Microflow QoS is enabled globally

QoS global counters:
  Total packets: 544393
  IP shortcut packets: 1410
  Packets dropped by policing: 0
  IP packets with TOS changed by policing: 467
  IP packets with COS changed by policing: 59998
  Non-IP packets with COS changed by policing: 0

Router#
```

Configuring Classification

This section contains information for configuring classification for the OSM QoS features.

Classification is the selection of traffic for QoS. OSMs classify IP traffic based on the packet IP precedence or IP DSCP value. OSMs classify MPLS traffic based on the MPLS EXP value in the topmost label in the MPLS label stack. Use the **class-map** command in global configuration mode to specify the name of the class map and define the class-matching criteria.

Restrictions and Usage Guidelines

The classification restrictions and usage guidelines are as follows:

- **Traffic types** —The classification information in this section is for IP and MPLS traffic. For information about configuring classification for EoMPLS and AToM traffic, refer to the “[How to Configure QoS with AToM](#)” section on page 11-78.
- **Traffic classes** —You can configure up to 64 discrete traffic classes in a single policy map, one class for each IP DSCP value. In addition to the traffic classes you specify, the class-default class is predefined when you create the policy map. It is the class to which traffic is directed if that traffic does not satisfy the match criteria of other classes defined in the policy map.
- **Using the match command** —Only the **match ip precedence**, **match ip dscp**, and **match mpls experimental** commands are supported. Access control lists and other criteria are not supported as match criteria for traffic classes.



Note

In 122-18.SXF7 and subsequent SXF releases, the **show policy-map interface** command does not display packet counters for policy-map classes containing no Actions.

Configuration Tasks

To configure classification, perform this task in global configuration mode on the Multilayer Switch Feature Card (MSFC):

	Command	Purpose
Step 1	Router(config)# class-map [match-all match-any] class-name	Creates a class map to be used for matching packets to a class.
Step 2	Router(config-cmap)# match [ip dscp <i>ip-dscp-value</i> ip precedence <i>ip-precedence-value</i> mpls experimental <i>mpls-exp-value</i>]	Configures a specific IP precedence, IP DSCP, or MPLS EXP value as a match criterion.

Configuring Class-Based Traffic Shaping

This section contains information for configuring class-based traffic shaping.

QoS on the OSMs supports both inbound and outbound class-based traffic shaping. Class-based traffic shaping on the OSMs limits the traffic to the configured rate. To configure class-based traffic shaping, use the **shape average** command.

Restrictions and Usage Guidelines

The class-based traffic shaping restrictions and usage guidelines are as follows:

- **Traffic shaping granularity**—On the OSMs, the granularity of the shaping rate is 1/255 of the link rate or the hierarchical shape rate. The OSMs automatically round the configured rate to the nearest multiple of 1/255. The **show policy-map interface** command displays the rounded shaping rate.



Note

For the Enhanced GE WAN, Enhanced OC3 POS, and Enhanced OC12 POS, the granularity of the shaping rate is 64,000 bps.

- **Minimum traffic shaping rate**—As shown in [Table 9-1](#), the **shape average** command has a minimum rate.
 - For OC-3c and above interfaces, the minimum rate is 1/255 of the physical interface speed.
 - For T3 and lower interfaces, the minimum rate is 256,000 bps.
 - For hierarchical shaped interfaces, the minimum rate of the parent policy is 1,000,000 bps and the minimum rate of the child policy is the greater of (a) 256,000 bps or (b) 1/255 of the parent shape rate.
- **Traffic shaping bursts**—The OSMs have a fixed burst size; therefore, the OSMs do not support the **shape average** command committed burst (Bc) and excess burst (Be) parameters. When you monitor a shaping policy with the **show policy interface** command, the output shows values for the Bc and Be parameters that the **show** command generates. The OSMs do not use these values.

Table 9-1 Minimum QoS Rates for shape average Command

Physical Interface	Minimum rate for shape average command (bps)
T3 and below	256,000
OC3 POS	607,000
Enhanced OC3 POS	256,000
OC12 POS	2,439,000
Enhanced OC12 POS	256,000
OC48 POS	9,756,000
GE WAN	1,000,000
Enhanced GE WAN	256,000
hierarchical traffic shaping (parent)	1,000,000
hierarchical traffic shaping (child)	Greater of (a) 256,000 bps or (b) 1/255 of the parent rate

Configuration Tasks

To configure class-based traffic shaping, use the Modular QoS CLI. Define the class of traffic with the **class-map** command as shown previously, create a policy map as shown previously that contains the **shape average** command, and apply the policy to the appropriate interface with the **service-policy** command.

To configure the class of traffic, see the [“Configuring Classification” section on page 9-3](#). To configure a policy with class-based traffic shaping and to assign the policy to an interface, perform the following tasks in global configuration mode.

**Note**

The following tasks assume that a traffic class has already been created.

	Command	Purpose
Step 1	Router(config)# policy-map <i>policy_name</i>	Specifies the name of the policy map to configure.
Step 2	Router(config-pmap)# class <i>class-name</i>	Specifies the name of a predefined class included in the service policy.
Step 3	Router(config-pmap-c)# shape average <i>cir</i> ^{1 2}	Shapes traffic to the indicated bit rate for the specified class.
Step 4	Router(config)# interface <i>interface-name</i>	Specifies the interface to which the policy map will be applied.
Step 5	Router(config-if)# service-policy [input output <i>policy-name</i>]	Attaches the specified policy map to the interface.

1. Only supported parameters are shown.
2. See [Table 9-1 on page 9-5](#).

Configuration Example

This example shows how to configure a low shape-rate queue and a high shape-rate queue:

- Step 1** Create traffic classes containing the match criteria for the two classes by defining the class maps as follows:

```
Router(config)# class-map match-all gold-data
Router(config-cmap)# match ip dscp 40
Router(config-cmap)# exit
Router(config)# class-map match-all bronze-data
Router(config-cmap)# match ip dscp 8
Router(config-cmap)# exit
```

- Step 2** Define a service policy to contain policy specifications for the two classes—gold-data and bronze-data. The match criteria for these classes are defined in [Step 1](#).

```
Router(config)# policy-map policy1
Router(config-pmap)# class gold-data
Router(config-pmap-c)# shape average 2000000
Router(config-pmap-c)# exit
Router(config-pmap)# class bronze-data
Router(config-pmap-c)# shape average 5000000
Router(config-pmap)# class class-default
Router(config-pmap-c)# shape average 3000000
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)#
```

- Step 3** Apply the policy map to the appropriate interface. The following example attaches the policy as an output policy:

```
Router(config)# interface pos7/1
Router(config-if)# service-policy output policy1
Router(config-if)# exit
```

- Step 4** Display the policy information for the interface:



Note The OSMs do not use the Bc and Be values in the output below. The CLI generates these values.

```
Router# show policy interface pos7/1
POS7/1

Service-policy output: policy1

Class-map: gold-data (match-all)
 795533 packets, 271276753 bytes
 30 second offered rate 17269000 bps, drop rate 2939000 bps
Match: ip dscp cs5
queue size 0, queue limit 128
packets output 660256, packet drops 135277
tail/random drops 135277, no buffer drops 0, other drops 0
shape (average) cir 20000000 bc 80000 be 80000
target shape rate 20000000
(shape parameter is rounded to 19,513,000 bps due to granularity)

Class-map: bronze-data (match-all)
 795533 packets, 271276753 bytes
 30 second offered rate 17269000 bps, drop rate 13687000 bps
Match: ip dscp cs1
queue size 0, queue limit 128
packets output 165164, packet drops 630369
tail/random drops 630369, no buffer drops 0, other drops 0
shape (average) cir 5000000 bc 20000 be 20000
target shape rate 5000000
(shape parameter is rounded to 4,878,000 bps due to granularity)

Class-map: class-default (match-any)
 3182121 packets, 1085103261 bytes
 30 second offered rate 69075000 bps, drop rate 47581000 bps
Match: any
queue size 0, queue limit 128
packets output 990320, packet drops 2191801
tail/random drops 2191801, no buffer drops 0, other drops 0
shape (average) cir 30000000 bc 120000 be 120000
target shape rate 30000000
(shape parameter is rounded to 29,270,000 bps due to granularity)
```



Note For information on hierarchical traffic shaping policies, see the [“Configuring Hierarchical Traffic Shaping”](#) section on page 9-17.

Configuring Class-Based Weighted Fair Queuing

This section contains information for configuring Class-Based Weighted Fair Queueing (CBWFQ). CBWFQ provides guaranteed bandwidth rate to a non-priority class. Under congestion conditions, the class receives the guaranteed bandwidth. To configure CBWFQ, use the **bandwidth** command.

**Note**

Low Latency Queueing (LLQ) provides guaranteed bandwidth for the priority classes. The sum of all bandwidth on a link guaranteed by CBWFQ for non-priority classes and LLQ for priority classes cannot exceed 99% of the total available link bandwidth. For more information on LLQ, see the “[Configuring Low Latency Queueing](#)” section on page 9-12.

Restrictions and Usage Guidelines

The CBWFQ restrictions and usage guidelines are as follows:

- **OSM support**— CBWFQ is not supported in DPT mode. CBWFQ is supported in POS mode on all OSMs except the OSM-2OC48/1DPT and OSM-4GE-WAN-GBIC.
- **Bandwidth granularity**—On the OSMs, the granularity of the CBWFQ rate is 1/255 of the link rate or the hierarchical shape rate. The OSMs automatically round the configured rate to the nearest multiple of 1/255. The **show policy-map interface** command displays the rounded CBWFQ rate.
- **Minimum bandwidth rate**—On the OSMs, the minimum CBWFQ rate is the greater of (a) 256 Kbps or (b) 1% of the link rate or the hierarchical shape rate.
- **Bandwidth allocation**—When a link is not experiencing congested conditions, the unused (or excess) bandwidth is shared among all classes. The excess bandwidth available to a class is in proportion to its guaranteed bandwidth specified by the **priority** or **bandwidth** commands. For example, if one class is guaranteed 20% of the link and a second class is guaranteed 10% of the link, then the first class receives twice as much excess bandwidth as the second class.
- **Using class-default**—The guaranteed bandwidth of the class-default class is by default equal to the link bandwidth minus the guaranteed bandwidth allocated to the user-defined traffic classes (with the **bandwidth** and **priority** commands). At least 1% of the link bandwidth is always reserved for the default traffic class. You can alter the bandwidth allocated to the default traffic by using the **bandwidth** command with the class-default class.
- Because the MSFC and PFC do not support CBWFQ, configuring CBWFQ on a system configured with a Supervisor Engine 720 and an OSM-1CHOC12/T1-SI, might cause the output of the show policy-map interface command to display a packet counter of 0 for a serial interface.
- When you configure a Class Based Weighted Fair Queueing or Low Latency Queueing in combination with any Feature requiring MSFC/PFC on a OSM-1CHOC12/T1-SI, the counters for **show policy-map interface** does not increment. This is because some of the features require MSFC/PFC processing, and MSFC/PFC does not support WAN Class Based Weighted Fair Queueing or Low Latency Queueing and the packets are not accounted for QoS. Examples of such Features are:
 - Frame-relay ip tcp header-compression
 - Frame-relay ip rtp header-compression
 - Access-list 101 permit ip any any log

**Note**

MSFC/PFC processes the **log** keyword in the access-lists results of a packet.

Configuration Tasks

These sections describe the configuration tasks for CBWFQ:

- [Configuring a Service Policy in the Policy Map, page 9-9](#)

- [Displaying the CBWFQ Configuration and Statistics, page 9-9](#)

Configuring a Service Policy in the Policy Map

To configure CBWFQ, use the Modular QoS CLI. Define the class of traffic with the **class-map** command, create a policy map that contains the **bandwidth** command, and apply the policy to the appropriate interface with the **service-policy** command.

To configure the class of traffic, see the “[Configuring Classification](#)” section on page 9-3. To configure a policy with CBWFQ and assign the policy to an interface, perform the following tasks in global configuration mode:

	Command	Purpose
Step 1	Router(config)# policy-map <i>policy-map</i>	Specifies the name of the policy map to configure.
Step 2	Router(config-pmap)# class <i>class-name</i>	Specifies the name of a predefined class included in the service policy.
Step 3	Router(config-pmap-c)# bandwidth <i>bandwidth-kbps</i> percent % <i>of available bandwidth</i> ^{1 2}	Specifies the percentage of available bandwidth in kilobits per second to be assigned to packets that meet the match criteria of the associated traffic class.
Step 4	Router(config)# interface <i>interface-name</i>	Specifies the interface to which the policy map will be applied.
Step 5	Router(config-if)# service-policy [output <i>policy-name</i>]	Attaches the specified policy map to the interface.

1. Only the parameters shown are supported.
2. The **bandwidth** command has a minimum rate that is one percent of the physical interface speed or the hierarchical shaped rate. See [Table 9-1](#).

Displaying the CBWFQ Configuration and Statistics

Use the commands below to display the configuration of a service policy and its associated traffic classes:

Command	Purpose
Router# show policy-map	Displays all configured service policies.
Router# show policy-map <i>policy-map-name</i>	Displays the user-specified service policy.
Router# show policy-map interface	Displays statistics and configurations of all input and output policies, which are attached to an interface.
Router# show policy-map interface <i>interface-spec</i>	Displays configuration and statistics of the input and output policies attached to a particular interface.
Router# show policy-map interface <i>interface-spec</i> output	Displays configuration statistics of the output policy attached to an interface.
Router# show policy-map [interface [<i>interface-spec</i> [output] [class <i>class-name</i>]]	Displays the configuration and statistics for the class name configured in the policy.

This example shows the information displayed when you enter the **show policy-map interface** command:

```
Router1-PE# show policy-map interface

POS6/2
service-policy output:s

queue stats for all priority classes:
  queue size 0, queue limit 32655
  packets output 0, packet drops 0
  tail/random drops 0, no buffer drops 0, other drops 0

class-map:dscp0 (match-all)
  0 packets, 0 bytes
  30 second offered rate 0 bps, drop rate 0 bps
  match:ip dscp 0
  queue size 0, queue limit 610
  packets output 0, packet drops 0
  tail/random drops 0, no buffer drops 0, other drops 0
  shape:cir 2440000, Bc 9760, Be 9760
  (shape parameter is rounded to 2,439,000 bps due to granularity)
  output bytes 0, shape rate 0 bps

class-map:dscp1 (match-any)
  0 packets, 0 bytes
  30 second offered rate 0 bps, drop rate 0 bps
  match:ip dscp 1
    0 packets, 0 bytes
    30 second rate 0 bps
  queue size 0, queue limit 100000
  packets output 0, packet drops 0
  tail/random drops 0, no buffer drops 0, other drops 0
  bandwidth:kbps 400000, weight 64
  (bandwidth parameter is rounded to 397,592 kbps due to granularity)

class-map:dscp2 (match-all)
  0 packets, 0 bytes
  30 second offered rate 0 bps, drop rate 0 bps
  match:ip dscp 2
  Priority:21% (130620 kbps), burst bytes 3265500, b/w exceed drops:0
  (Priority parameter is rounded to 129,278 kbps due to granularity)

class-map:class-default (match-any)
  0 packets, 0 bytes
  30 second offered rate 0 bps, drop rate 0 bps
  match:any
    0 packets, 0 bytes
    30 second rate 0 bps
  queue size 0, queue limit 11422
  packets output 0, packet drops 0
  tail/random drops 0, no buffer drops 0, other drops 0
```

Configuration Example

This example shows how to configure traffic classes, create a service policy, and attach it to an interface:

- Step 1** Two traffic classes are created and their match criteria are defined. For the first traffic class, called class1, DSCP 30 is used as the match criterion. For the second traffic class, called class2, DSCP 10 is used as the match criterion. Packets are checked against the contents of these criteria to determine if they belong to the traffic class.

```
Router(config)# class-map class1
Router(config-cmap)# match ip dscp 30
Router(config-cmap)# exit
```

```
Router(config)# class-map class2
Router(config-cmap)# match ip dscp 10
Router(config-cmap)# exit
```

- Step 2** A service policy called policy1 is defined to associate QoS features with the two traffic classes—class1 and class2. The match criteria for these traffic classes were defined in [Step 1](#).

```
Router(config)# policy-map policy1
Router(config-pmap)# class class1
Router(config-pmap-c)# bandwidth 30000
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# class class2
Router(config-pmap-c)# bandwidth 20000
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)#
```

- Step 3** After you define a service policy, you can attach it to one or more interfaces to specify a service policy for those interfaces. Although you can assign the same service policy to multiple interfaces, each interface can have only one service policy attached at the input and one policy map attached at the output at one time.

```
Router(config)# interface pos 2/0
Router(config-if)# service-policy output policy1
Router(config-if)# exit
```

In the following example, CBWFQ is configured on an OC-3 link. Class foo and class bar are guaranteed minimum bandwidth of 20 percent and 30 percent of link rate, respectively. The remaining bandwidth is equally distributed between class-default and class baz, each gets 25 percent of link rate. However, because class baz is shaped to receive a maximum of 15 Mbps (or 10 percent of the link rate), the bandwidth allocated to class baz is capped by this specified shape value, and class baz is guaranteed a minimum bandwidth of 15 Mbps.

```
class-map foo
  match ip dscp 10
class-map bar
  match ip dscp 20
class-map baz
  match ip dscp 30

policy-map foobar
  class foo
    bandwidth percent 20%
  class bar
    bandwidth percent 30%
  class baz
    shape average 15000000

int pos2/0
```

```
service-policy output foobar
```

Configuring Low Latency Queuing

Low latency queuing (LLQ) lets you specify low-latency behavior for a traffic class. LLQ allows delay-sensitive data to be given preferential treatment. You can give one or more classes priority status. You configure LLQ with the **priority** command.

The **priority** command configures guaranteed bandwidth to a priority class under worst-case congestion scenarios.

Restrictions and Usage Guidelines

The LLQ restrictions and usage guidelines are as follows:

- LLQ is not supported in the input direction.
- CBWFQ and LLQ both provide guaranteed bandwidth for their respective classes. The sum of all bandwidth on a link guaranteed by CBWFQ for non-priority classes and LLQ for priority classes cannot exceed 99% of the total available link bandwidth. Additionally, Cisco recommends that the sum of all the LLQ classes bandwidth remain below 25% of the main interface bandwidth. For more information on CBWFQ, see the [“Configuring Class-Based Weighted Fair Queuing”](#) section on page 9-7.
- **OSM support**—LLQ is supported in POS mode on the 2-port OC-48c/STM-16 POS/DPT OSMs; it is not supported in DPT mode. CBWFQ is not supported on the OSM-4GE-WAN-GBIC.
- **Bandwidth granularity**—On the OSMs, the granularity of the priority rate is 1/255 of the link rate or the hierarchical shape rate. The OSMs automatically round the configured rate to the nearest multiple of 1/255. The **show policy-map interface** command displays the rounded priority rate.
- **Minimum priority rate**—As shown in [Table 9-2](#), the **priority** command has a minimum rate. For OC-3c and above interfaces, the minimum rate is 1/255 of the physical interface speed. For T3 and lower interfaces, the minimum rate is 256,000 bps.
- **Bandwidth allocation**—When a link is not under congested conditions, the unused (or excess) bandwidth is shared among all classes. The excess bandwidth available to a class is in proportion to its guaranteed bandwidth specified by the **priority** or **bandwidth** commands. For example, if one class is guaranteed 20% of the link and a second class is guaranteed 10% of the link, then the first class receives twice as much excess bandwidth as the second class.
- **LLQ burst**—The OSMs have a fixed burst size. They do not support the **priority** command burst parameter.
- **Bandwidth command**—You cannot configure the **bandwidth** command for a priority class.
- Because the MSFC and PFC do not support LLQ, configuring LLQ on a system configured with a Supervisor Engine 720 and an OSM-1CHOC12/T1-SI, might cause the output of the show policy-map interface command to display a packet counter of 0 for a serial interface.

Table 9-2 Minimum QoS Rates for priority Command

Physical Interface	Minimum Rate for priority Command (kbps)
T3 and below	256
OC3 POS	607

Table 9-2 Minimum QoS Rates for priority Command

Physical Interface	Minimum Rate for priority Command (kbps)
OC12 POS	2439
OC48 POS	9756
GE WAN	Not supported
Enhanced GE WAN	3921
Hierarchical traffic shaping (child)	Greater of (a) 256 or (b) 1/255 of the parent rate.

**Note**

Starting with Cisco IOS Release 12.2(18)SXE, the Low Latency Queuing feature is change for the POS OSMs and the OSM-2+4GE-WAN OSM. For further information see [Configuring Strict Priority LLQ Support on POS Optical Service Modules, page 3-29](#), and [Configuring Strict Priority Low Latency Queuing \(LLQ\) Support on the OSM-2+4GE-WAN+, page 4-4](#).

Configuration Tasks

To configure LLQ, use the Modular QoS CLI. Define the class of traffic with the **class-map** command, create a policy map that contains the **priority** command, and apply the policy to the appropriate interface with the **service-policy** command.

To configure the class of traffic, see the “[Configuring Classification](#)” section on page 3. To configure a policy with LLQ and to assign the policy to an interface, perform the following tasks in global configuration mode:

	Command	Purpose
Step 1	Router(config)# policy-map <i>policy-name</i>	Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy.
Step 2	Router(config-pmap)# class <i>class-name</i>	Specifies the name of a predefined class included in the service policy.
Step 3	Router(config-pmap-c)# priority <i>bandwidth-kbps</i> percent % of available bandwidth ¹ ²	Reserves a priority queue for CBWFQ traffic.
Step 4	Router(config)# interface <i>interface-name</i>	Specifies the interface to which the policy map will be applied.
Step 5	Router(config-if)# service-policy [<i>output policy-name</i>]	Attaches the specified policy map to the interface.

1. Only the parameters shown are supported.
2. See [Table 9-1 on page 9-5](#) and [Table 9-2 on page 9-12](#).

Configuration Example

This example shows how to configure a priority queue (with a guaranteed allowed bandwidth of 50 Mbps) reserved for traffic with an IP DSCP value of 40:

Step 1 Create traffic classes and specify match criteria by defining class maps.

```
Router(config)# class-map gold-data
Router(config-cmap)# match-any ip dscp 40
Router(config-cmap)# exit
Router(config)# class-map match bar
Router(config-cmap)# match-any ip dscp 8
Router(config-cmap)# exit
Router(config)#
```

Step 2 Create the policy map. In the example, a priority queue for the class gold-data is reserved with a guaranteed allowed bandwidth of 50 Mbps, and a bandwidth of 20 Mbps is configured for the class bar. The **service-policy** command attaches the policy map to interface pos 4/1.

```
Router(config)# policy-map policy1
Router(config-pmap)# class gold-data
Router(config-pmap-c)# priority 50000
Router(config-pmap)# class bar
Router(config-pmap-c)# bandwidth 20000
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# interface pos 4/1
Router(config-if)# service-policy output policy1
Router(config-if)# exit
```

Configuring Weighted Random Early Detection

Weighted Random Early Detection (WRED) is a congestion-avoidance mechanism that takes advantage of the congestion-control mechanism of TCP. By selectively dropping packets based on IP precedence prior to periods of high congestion, WRED tells the packet source to decrease its transmission rate. Edge routers assign IP precedence to packets as they enter the network. WRED is useful on any output interface where you expect to have congestion. However, WRED is usually used in the core routers of a network rather than at the edge. WRED uses these precedences to determine how it treats different types of traffic.

When a packet arrives, the average queue size is calculated and one of the following events occurs:

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

Restrictions and Usage Guidelines

The WRED restrictions and usage guidelines are as follows:

- **OSM support**—WRED is only supported on Enhanced OSMs.
- WRED is not supported in the input direction.
- In the case of a user-defined class, random-detect is supported in association with shaping or bandwidth only.
- WRED is supported for DSCP and EXP. For DSCP, you configure WRED with the **random-detect dscp-based** command.
- IPv6 traffic for WRED configuration on OSMs are accounted in class-default within WRED as OSMs do not support WRED states for IPv6.

For more details on the queue calculations and how WRED works, refer to the section “About WRED” in the chapter “Congestion Avoidance Overview” in the *Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.2* at the following URL:

http://www.cisco.com/en/US/docs/ios/12_2/qos/configuration/guide/fqos_c.html

Configuration Tasks

To configure WRED, use the Modular QoS CLI. Define the class of traffic with the **class-map** command, create a policy map that contains the **random-detect** command, and apply the policy to the appropriate interface with the **service-policy** command.

To configure the class of traffic, see the “[Configuring Classification](#)” section on page 9-3. To configure a policy with WRED and to assign the policy to an interface, perform the following tasks in global configuration mode:

	Command	Purpose
Step 1	Router(config)# policy-map <i>policy-name</i>	Specifies the name of the policy map to configure.
Step 2	Router(config-pmap)# class <i>class-name</i>	Specifies the name of a predefined class included in the service policy.
Step 3	Router(config-pmap-c)# random-detect	Enables WRED.
Step 4	Router(config)# interface <i>interface-name</i>	Specifies the interface to which the policy map will be applied.
Step 5	Router(config-if)# service-policy [<i>output policy-name</i>]	Attaches the specified policy map to the interface.

Configuration Example

This example shows how to configure WRED.

```
Router# config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# policy-map wred_test
Router(config-pmap)# class class-default
Router(config-pmap-c)# random-detect
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# interface pos 7/1
Router(config-if)# service-policy output wred_test
Router(config-if)# end
Router# show policy-map interface pos 7/1
POS7/1
```

Service-policy output: wred_test

```
Class-map: class-default (match-any)
 16634097 packets, 8217243918 bytes
 30 second offered rate 482198000 bps, drop rate 0 bps
Match: any
queue size 0, queue limit 128
packets output 16634097, packet drops 0
tail/random drops 0, no buffer drops 0, other drops 0
Random-detect:
  Exp-weight-constant: 3 (1/8)
  Mean queue depth: 0
  Class Random      Tail   Minimum  Maximum  Mark      Output
   drop            drop threshold threshold probability packets
0      104806          0      32       64      1/10     3026812
1      104569          0      36       64      1/10     3027050
2      104732          0      40       64      1/10     3026884
3      104169          0      44       64      1/10     3027449
4      103047          0      48       64      1/10     3028569
5      103156          0      52       64      1/10     3028460
```


6	0	0	56	64	1/10	0
7	0	0	60	64	1/10	0

Configuring Hierarchical Traffic Shaping

Hierarchical traffic shaping allows multiple classes of traffic to be shaped at a single rate. A hierarchical traffic shaping policy consists of a child policy that identifies one or more classes of traffic, and a parent policy that shapes the output of the traffic classes into a single shape rate. You can apply a nested policy to an interface or subinterface.

Restrictions and Usage Guidelines

The hierarchical traffic shaping restrictions and usage guidelines are as follows:

- Hierarchical traffic shaping is supported as an output policy only.
- Hierarchical traffic shaping is supported on the interfaces and subinterfaces for the OSM-2+4GE-WAN OSM.



Note Shape average is supported on the OSMs; shape peak is not supported on the OSMs.

- For the OSM-2+4GE-WAN OSM and enhanced POS OSMs starting with Cisco IOS release 12.2(18)SX, the **police** command is not supported in a child class unless it is coupled with the **priority** command in the same child class.
- Hierarchical traffic shaping is supported on Frame Relay encapsulated subinterfaces on the original and enhanced POS OSMs and in POS mode on the DPT OSM.
- Hierarchical traffic shaping is supported on Frame Relay on encapsulated main interfaces using map class on enhanced POS OSMs.
- Hierarchical traffic shapping is supported on PPP encapsulated interfaces on the enhanced POS OSMs and in POS mode on the DPT OSM.



Note If a WAN interface on an OSM already has a hierarchical MQC policy map attached, do not attempt to convert it to a flat policy map. Conversely, do not attempt to convert a flat policy map to a hierarchical MQC policy map unless you first remove the policy from the attached interfaces before you convert the policy.

Configuration Task

To configure hierarchical traffic shaping, use the Modular QoS CLI to define the parent and child policies. For the child policy, define the classes of traffic with the **class-map** command and create a policy with the **policy-map** command. For the parent policy, create a policy with the **policy-map** command, apply the child policy to the default class with the **service-policy** command, and apply the parent policy to the appropriate interface with the **service-policy** command.

To configure the classes of traffic, see the “[Configuring Classification](#)” section on page 9-3. To configure the child and parent policies for hierarchical traffic shaping, perform the following tasks in global configuration mode:

	Command	Purpose
Step 1	Router(config)# policy-map <i>child-policy-name</i>	Specifies the name of the child policy map to configure.
Step 2	Router(config-pmap)# class <i>class-name</i>	Specifies the name of a predefined class included in the service policy.
Step 3	Router(config-pmap-c)# priority <i>bandwidth-kbps</i> percent % of available bandwidth ¹	Gives priority to a class of traffic belonging to the policy map.
Step 4	Router(config-pmap)# class <i>class-name</i>	Specifies the name of a predefined class included in the service policy.
Step 5	Router(config-pmap-c)# bandwidth <i>bandwidth-kbps</i> percent % of available bandwidth ²	Specifies the percentage of available bandwidth in kilobits per second to be assigned to packets that meet the match criteria of the associated traffic class.
Step 6	Router(config)# policy-map <i>parent-policy-name</i>	Specifies the name of the parent policy map to configure.
Step 7	Router(config-pmap)# class <i>class-name</i>	Specifies the name of a predefined class included in the service policy.
Step 8	Router(config-pmap-c)# shape average cir	Shapes traffic to the indicated bit rate for the specified class.
Step 9	Router(config-pmap-c)# service-policy <i>child-policy-name</i>	Links the parent and child policy and class.
Step 10	Router(config)# interface <i>interface-name</i>	Specifies the interface to which the policy map will be applied.
Step 11	Router(config-if)# service-policy [output <i>parent-policy-name</i>]	Attaches the specified nested parent and child policies to the interface.

1. Only the parameters shown are supported.
2. Only the parameters shown are supported.

Configuration Example

This example shows a nested traffic policy configuration where traffic matching the class called voice will be guaranteed 3200 Kbps, or 10% of parent_policy's shape average:

```
Router(config)# class-map match-all voice
Router(config-cmap)# match ip dscp 5
Router(config-cmap)# exit
```

```
Router(config)# policy-map child_policy
Router(config-pmap)# class voice
Router(config-pmap-c)# priority percent 10
Router(config-pmap-c)# exit
Router(config-pmap)# exit

Router(config)# policy-map parent_policy
Router(config-pmap)# class class-default
Router(config-pmap-c)# shape average 32000000
Router(config-pmap-c)# service-policy child_policy
Router(config-pmap-c)# exit
Router(config-pmap)# exit

Router(config)# interface Serial6/1:1.1 point-to-point
Router(config-subif)# service-policy parent_policy
```

Configuring Queue Limit

For the class-based traffic shaping and CBWFQ features, you can specify or modify the maximum number of packets the queue can hold for a class policy configured in a policy map using the **queue-limit** command from policy-map class configuration mode.

Restrictions and Usage Guidelines

The queue limit restrictions and usage guidelines are as follows:

- **OSM queue-limit values**—The default queue-limit value is chosen as a function of the OSM card type and the link encapsulation configured for an interface. It is not chosen based on the bandwidth assigned to the traffic class or the amount of buffer memory. You can change the default queue limit to one of the following values: 18, 25, 42, 128, 256, 512, 1024, 2048, 4096, 8192, 16384, or 32768.



Note If you use queue-limit values other than the default values, the value is rounded down. For example, a queue-limit value of 3000 is rounded to 2048.

- In the case of a user-defined class, queue-limit is supported in association with shaping and bandwidth only.
- Flat policy maps with CBWFQ classes on subinterfaces with smaller packet sizes (less than 128 packets) may experience tail drops because of the default queue size (128 packets) (CSCeg73678). For these classes, if the tail drop is unacceptable, use the queue-limit to adjust the queue size to a larger value.

Configuring Tasks

To configure the queue-limit, use the Modular QoS CLI. Define the class of traffic with the **class-map** command, create a policy map that contains the **queue-limit** command, and apply the policy to the appropriate interface with the **service-policy** command.

To configure the class of traffic, see the “[Configuring Classification](#)” section on page 9-3. To configure a policy with queue-limit and to assign the policy to an interface, perform the following tasks in global configuration mode:

	Command	Purpose
Step 1	Router(config)# policy-map <i>policy-name</i>	Specifies the name of the policy map to configure.
Step 2	Router(config-pmap)# class <i>class-name</i>	Specifies the name of a predefined class included in the service policy.
Step 3	Router(config-if)# queue-limit <i>number-of-packets</i>	Specifies the maximum number of packets the queue can hold for a class policy configured in a policy map.
Step 4	Router(config)# interface <i>interface-name</i>	Specifies the interface to which the policy map will be applied.
Step 5	Router(config-if)# service-policy [<i>input</i> <i>output</i> <i>policy-name</i>]	Attaches the specified policy map to the interface.



Note

To remove the queue packet limit from a class, use the **no queue-limit** form of this command.

Configuration Example

In the following example, a policy map called policy11 is configured to contain policy for a class called cls203, and the queue limit is set to 42 packets.

```
Router(config)# policy-map policy11
Router(config-pmap)# class cls203
Router(config-pmap-c)# queue-limit 42
```

Configuring QoS: Match VLAN

The QoS: Match VLAN feature provides trunk VLAN match and classification in Modular QoS Command-Line Interface (MQC) class maps for the OSM-2+4GE-WAN+ interface.



Note

This feature was first supported in Cisco IOS Release 12.2(18)SXE.

Classification based on the **match vlan** command is supported by the following QoS features:

- Ingress/egress shaping
- Egress class based weighted fair queuing (CBWFQ)
- Egress low latency queuing (LLQ Strict Priority)
- Egress weighted random early detection (WRED)

- Hierarchical QoS on non-default class

Restrictions and Usage Guidelines

- In a hierarchical policy, match VLAN classes in a child policy are not supported. For example, the following configuration is not supported:

```
!
policy not-supported-policy
  class class-default
    shape average 100000000
    service-policy child-m-vlan

Policy Map child-m-vlan
  class vlan150
  bandwidth 20 (%)
!
```

- This feature supports the **class-map match-all** and **match-any** parameters.
- A match VLAN policy is always applied to a main interface.

Configuration Tasks

To configure QoS: Match VLAN support, perform the following tasks, beginning in global configuration mode:

	Command or Action	Purpose
Step 1	Router(config)# class-map <i>class-map-name</i> Example: Router(config)# class-map vlan100	Specifies the name of the class map to be created or modified.
Step 2	Router(config-cmap)# match vlan <i>vlan-id</i> Example: Router(config)# match vlan 100	Specifies the VLAN ID used as match criteria. The valid value range is 1 to 4095. Alternately, a range of VLAN IDs can be used as match criteria. Use a hyphen to separate the VLAN IDs in each range. Use a space to separate each range of VLANs.

Configuration Examples

The following example shows a sample hierarchical match VLAN policy:

```
Router# configure terminal
Router(config)# policy-map vlan-pol-example
Router(config-pmap)# class vlan150
Router(config-pmap-c)# shape average 200000000 800000 800000
Router(config-pmap-c)# service-policy child
Router(config-pmap-c)# exit
Router(config-pmap)# class vlan170
Router(config-pmap-c)# shape average 200000000 800000 800000
Router(config-pmap-c)# service-policy child
!
Router(config)# policy-map child
Router(config-pmap)# Class prec2
```

```

Router(config-pmap-c)# bandwidth 20 (%)
Router(config-pmap)# Class prec3
Router(config-pmap-c)# bandwidth 30 (%)
Router(config-pmap)# Class prec1
Router(config-pmap-c)# bandwidth 10 (%)

```

The following example creates a class map for multiple VLANs, creates a policy map for shaping, and attaches an egress interface:

```

Router# configure terminal
Router(config)# class-map vlan_multi
Router(config-cmap)# match vlan 100-102 200 300
Router(config-cmap)# exit
Router(config)# policy-map pol_multi_vlan
Router(config-pmap)# class vlan_multi
Router(config-pmap-c)# shape average 100000000
Router(config-pmap-c)# exit
Router(config)# interface ge7/1.100
Router(config-subif)# encapsulation dot1q 100
Router(config-subif)# no shut
Router(config-int)# interface ge7/1.101
Router(config-subif)# encapsulation dot1q 101
Router(config-subif)# no shut
Router(config-int)# interface ge7/1.102
Router(config-subif)# encapsulation dot1q 102
Router(config-subif)# no shut
Router(config-int)# interface ge7/1.200
Router(config-subif)# encapsulation dot1q 200
Router(config-subif)# no shut
Router(config-int)# interface ge7/1.300
Router(config-subif)# encapsulation dot1q 300
Router(config-subif)# no shut
Router(config-subif)# exit
Router(config)# interface ge7/1
Router(config-int)# service-policy output pol_multi_vlan
Router(config-int)# no shut

```

Distribution of Remaining Bandwidth

You can use MQC to specify how the "remaining" bandwidth is distributed among the output queues on a Cisco 7600 router interface or subinterface. "Remaining" bandwidth is the available bandwidth left on an interface or subinterface after all guaranteed traffic is accounted for. The amount of remaining bandwidth available for use is determined by the excess information rate (EIR) configured for the queue.

In MQC, the **bandwidth remaining percent** command allows you to configure the remaining bandwidth for output queues. See the "bandwidth remaining percent" section for more information.

The following example shows how to use the **bandwidth remaining percent** command to distribute percentages of remaining bandwidth to various traffic classes in a policy map.

```

Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# policy-map myPolicy
Router(config-pmap)# class class-default
Router(config-pmap-c)# bandwidth remaining percent 20
Router(config-pmap-c)# class prec1
Router(config-pmap-c)# bandwidth remaining percent 30
Router(config-pmap-c)# class prec2
Router(config-pmap-c)# bandwidth remaining percent 10
Router(config-pmap-c)# bandwidth percent 50
Router(config-pmap-c)# ^Z

```

```

Router#
20:44:36: %SYS-5-CONFIG_I: Configured from console by console
Router# show policy-map myPolicy
  Policy Map myPolicy
    Class prec1
      bandwidth remaining percent 30
    Class prec2
      bandwidth percent 50
      bandwidth remaining percent 10
    Class class-default
      bandwidth remaining percent 20

```

Command Reference

To specify how the "remaining" bandwidth is distributed among the output queues on a Cisco 7600 series router interface or subinterface, use the MQC **bandwidth remaining percent** command in policy-map class configuration mode. To remove the percentage of remaining bandwidth specified for a traffic class, use the **no** form of this command.

bandwidth remaining percent *percentage*

no bandwidth remaining percent *percentage*

Syntax Description

<i>percentage</i>	Specifies a percentage value for the amount of guaranteed bandwidth, based on a relative percent of available bandwidth, to be assigned to the class. The percentage can be a number between 1 and 99.
-------------------	--

Defaults

This command has no default behavior or values.

Command Modes

Modular QoS policy-map class configuration

Command History

12.2(18)SXE	This command was introduced on the Cisco 7600 series router.
-------------	--

Usage Guidelines

Use the MQC bandwidth remaining percent command to specify how the "remaining" bandwidth is distributed among the output queues on a Cisco 7600 series router interface or subinterface. "Remaining" bandwidth is the available bandwidth left on an interface or subinterface after all guaranteed traffic is accounted for.

The bandwidth remaining percent command allows you to configure the remaining bandwidth for output queues. The percentage parameter specified with the bandwidth remaining percent command is translated into an internal excess information rate (EIR) value between 0 and 255. The aggregate of all user-configured EIR bandwidth percentages cannot exceed 100 percent.

If the aggregate of all remaining bandwidth is less than 100 percent, the remainder is evenly split among user queues (including the default queue) that do not have a remaining bandwidth percentage configured. The minimum EIR value of each output queue is 1.

The EIR parameter for the network control queue is fixed at 128 and is not configurable.

If you have not configured a committed information rate (CIR) value for the default queue and it is the only user queue, the default queue receives half of the remaining bandwidth percentage of the network control queue.

Unsupported Frame Relay-Specific QoS Features

The following Frame-Relay specific QoS features are not supported:

- Adaptive traffic shaping
- Adaptive policing
- DLCI priority levels
- DE bit support
- The **fair-queue** command and **random-detect** command on the main interface

Cisco IPv6 QoS on the OSMs

Cisco IPv6 QoS on the OSMs supports all QoS features supported with IPv4 QoS. For general information on Cisco IPv6 QoS, see *Implementing QoS for IPv6 for Cisco IOS Software* at:

<http://www.cisco.com/en/US/docs/ios/ipv6/configuration/guide/ip6-qos.html>

Restrictions and Usage Guidelines

The Cisco IPv6 QoS restrictions and usage guidelines are as follows:

- The **match precedence** or **match dscp** command match both IPv4 and IPv6 traffic based upon the QoS marking in the packet Layer 3 header.
- In conjunction with the **match precedence** or **match dscp** command, you can specify the **match protocol ipv6** command in the class map to match only IPv6 precedence or DSCP.
- In conjunction with the **match precedence** or **match dscp** command, you can specify the **match protocol ip** command in the class map to match only IPv4 precedence or DSCP.
- IPv6 QOS is supported only on Enhanced OSMs.
- There can only be one **match protocol** command per class map.