



MPLS Quality of Service

The MPLS Quality of Service feature (formerly named as the MPLS CoS feature) enables you to provide differentiated services across an MPLS network. To satisfy a wide range of networking requirements, you can specify the class of service applicable to each transmitted IP packet. Different classes of service can be established for IP packets by setting the IP precedence bit in the header of each packet.

- [Prerequisites for MPLS Quality of Service, page 1](#)
- [Information About MPLS Quality of Service, page 2](#)
- [How to Configure MPLS Quality of Service, page 7](#)
- [Configuration Examples for MPLS Quality of Service, page 14](#)
- [Additional References for MPLS Quality of Service, page 19](#)
- [Feature Information for MPLS Quality of Service, page 20](#)

Prerequisites for MPLS Quality of Service

To use MPLS CoS to full advantage in your network, the following functionality must be supported:

- Multiprotocol Label Switching (MPLS)—MPLS is the standardized label switching protocol defined by the Internet Engineering Task Force (IETF).
- Cisco Express Forwarding—Cisco Express Forwarding is an advanced Layer 3 IP switching technology that optimizes performance and scalability in networks that handle large volumes of traffic and that exhibit dynamic traffic patterns.
- Asynchronous Transfer Mode (ATM)—ATM signaling support is required if you are using ATM interfaces in your network.

If you are using only packet interfaces in your network, ATM functionality is not needed.

- QoS features:
 - Weighted fair queueing (WFQ)—Used on non-GSR platforms, WFQ is a dynamic scheduling method that allocates bandwidth fairly to all network traffic.

WFQ applies priorities, or weights, to traffic to classify the traffic into flows and determine how much bandwidth to allow each flow. WFQ moves interactive traffic to the front of a queue to reduce response time and fairly shares the remaining bandwidth among high-bandwidth flows.

- Weighted random early detection (WRED)—WRED is a congestion avoidance mechanism that extends RED functionality by allowing different RED parameters to be configured per IP precedence value.

IP precedence bits, contained in the type of service (ToS) octet in the IP packet header, are used to denote the relative importance or priority of an IP packet. WRED uses these IP precedence values to classify packets into different discard priorities or classes of service.

- Modified deficit round robin (MDRR)—Used only on GSR platforms, MDRR is a traffic class prioritization mechanism that incorporates emission priority as a facet of quality of service. MDRR is similar in function to WFQ on non-GSR platforms.

In MDRR, IP traffic is mapped to different classes of service queues. A group of queues is assigned to each traffic destination. On the transmit side of the platform, a group of queues is defined on a per-interface basis; on the receive side of the platform, a group of queues is defined on a per-destination basis. IP packets are then mapped to these queues, based on their IP precedence value.

These queues are serviced on a round-robin basis, except for a queue that has been defined to run in either of two ways: strict priority mode or alternate priority mode.

In strict priority mode, the high priority queue is serviced whenever it is not empty; this ensures the lowest possible delay for high priority traffic. In this mode, however, the possibility exists that other traffic might not be serviced for long periods of time if the high priority queue is consuming most of the available bandwidth.

In alternate priority mode, the traffic queues are serviced in turn, alternating between the high priority queue and the remaining queues.

- Committed access rate (CAR)—CAR is a QoS feature that limits the input or output transmission rate on an interface and classifies packets by setting the IP precedence value or the QoS group in the IP packet header.

Information About MPLS Quality of Service

MPLS Quality of Service Overview

MPLS CoS functionality enables network administrators to provide differentiated services across an MPLS network. Network administrators can satisfy a wide range of networking requirements by specifying the class of service applicable to each transmitted IP packet. Different classes of service can be established for IP packets by setting the IP precedence bit in the header of each packet.

MPLS CoS supports the following differentiated services in an MPLS network:

- Packet classification
- Congestion avoidance
- Congestion management

The table below describes the MPLS CoS services and functions.

Table 1: MPLS CoS Services and Functions

Service	CoS Function	Description
Packet classification	Committed access rate (CAR). Packets are classified at the edge of the network before labels are assigned.	CAR uses the type of service (ToS) bits in the IP header to classify packets according to input and output transmission rates. CAR is often configured on interfaces at the edge of a network in order to control traffic flowing into or out of the network. You can use CAR classification commands to classify or reclassify a packet.
Congestion avoidance	Weighted random early detection (WRED). Packet classes are differentiated based on drop probability.	WRED monitors network traffic to anticipate and prevent congestion at common network and internetwork bottlenecks. WRED can selectively discard lower priority traffic when an interface becomes congested; WRED can also provide differentiated performance characteristics for different classes of service.
Congestion management	Weighted fair queuing (WFQ) for non-GSR platform. Packet classes are differentiated based on bandwidth requirements and finite delay characteristics. Modified deficit round robin (MDRR) for GSR platforms.	WFQ is an automated scheduling system that ensures fair bandwidth allocation to all network traffic. WFQ uses weights (priorities) to determine how much bandwidth each class of traffic is allocated. MDRR, similar in function to WFQ for non-GSR platforms, is a traffic prioritization scheme that maps IP traffic to different classes of service queues, based on the IP precedence value of each packet. The queues are then serviced on a round-robin basis.

MPLS CoS enables you to duplicate Cisco IP CoS (Layer 3) features as closely as possible in MPLS devices, including label edge switch routers (edge LSRs) and label switch routers (LSRs). MPLS CoS functions map nearly one-for-one to IP CoS functions on all types of interfaces.

Tag Switching and MPLS Terminology

The table below lists the existing legacy tag switching terms and the new, equivalent Multiprotocol Label Switching (MPLS) IETF terms used in this document and other related Cisco publications.

Table 2: Tag Switching Terms and Equivalent MPLS Terms

Old Designation	New Designation
Tag switching	Multiprotocol Label Switching
Tag (short for tag switching)	MPLS
Tag (item or packet)	Label
TDP (Tag Distribution Protocol)	LDP (Label Distribution Protocol). Cisco TDP and LDP (MPLS Label Distribution Protocol) closely parallel each other in function, but differ in detail, such as message formats and the commands required to configure the respective protocols and to monitor their operation
Tag switched	Label switched
TFIB (tag forwarding information base)	LFIB (label forwarding information base)
TSR (tag switching router)	LSR (label switching router)
TVC (tag VC, tag virtual circuit)	LVC (label VC, label virtual circuit)
TSP (tag switch path)	LSP (label switch path)

Interfaces Supporting MPLS CoS Features

The table below lists the MPLS CoS features that are supported on packet interfaces.

Table 3: MPLS CoS Features Supported on Packet Interfaces

MPLS CoS Feature	Cisco 7200 Series	Cisco 7500 Series	Cisco 12000 Series GSR
Per-interface WRED	Yes	Yes	Yes
Per-interface, per-flow WFQ	Yes	Yes	No
Per-interface, per-class WFQ	No (supported in 12.1 and 12.1T)	Yes	N/A

MPLS CoS Feature	Cisco 7200 Series	Cisco 7500 Series	Cisco 12000 Series GSR
Per-interface MDRR	N/A	N/A	Yes

The table below lists the MPLS CoS features that are supported on ATM interface.

Table 4: MPLS CoS Features Supported on ATM Interfaces

MPLS CoS and ATM Cards	Cisco 7200 Series	Cisco 7500 Series	Cisco 12000 Series GSR
MPLS WRED: <ul style="list-style-type: none"> Per interface Per VC 	<ul style="list-style-type: none"> Yes, available on the ATM Lite port adapter (PA-A1). No, available on the ATM Deluxe port adapter (PA-A3). 	<ul style="list-style-type: none"> Yes, available on the ATM Lite port adapter (PA-A1). No, available on the ATM Deluxe port adapter (PA-A3). 	<ul style="list-style-type: none"> Yes No
MPLS MDRR: <ul style="list-style-type: none"> Per interface Per VC 	<ul style="list-style-type: none"> N/A N/A 	<ul style="list-style-type: none"> N/A N/A 	<ul style="list-style-type: none"> Yes No
MPLS WFQ: <ul style="list-style-type: none"> Per interface, WFQ Per interface, per-class WFQ 	<ul style="list-style-type: none"> Yes, available on the ATM Lite port adapter (PA-A1). No 	<ul style="list-style-type: none"> Yes, available on the ATM Lite port adapter (PA-A1). Yes 	<ul style="list-style-type: none"> No No

.

LSRs Used at the Edge of an MPLS Network

Label switching routers (LSRs) used at the edge of a Multiprotocol Label Switching (MPLS) network backbone are devices running MPLS software. The edge LSRs can be at the ingress or the egress of the network.

At the ingress of an MPLS network, devices process packets as follows:

- 1 IP packets enter the edge of the MPLS network at the edge LSR.
- 2 The edge LSR uses a classification mechanism such as the Modular Quality of Service Command-Line Interface (MQC) to classify incoming IP packets and set the IP precedence value. Alternatively, IP packets can be received with the IP precedence value already set.
- 3 For each packet, the device performs a lookup on the IP address to determine the next-hop LSR.

- 4 The appropriate label is inserted into the packet, and the IP precedence bits are copied into the MPLS EXP bits in the label header.
- 5 The labeled packets are forwarded to the appropriate output interface for processing.
- 6 The packets are differentiated by class according to one of the following:
 - Drop probability—Weighted random early detection (WRED)
 - Bandwidth allocation and delay—Class-based weighted fair queuing (CBWFQ)

In either case, LSRs enforce the defined differentiation by continuing to employ WRED or CBWFQ on every ingress device.

At the egress of an MPLS network, devices process packets as follows:

- 1 MPLS-labeled packets enter the edge LSR from the MPLS network backbone.
- 2 The MPLS labels are removed and IP packets may be (re)classified.
- 3 For each packet, the device performs a lookup on the IP address to determine the packet's destination and forwards the packet to the destination interface for processing.
- 4 The packets are differentiated by the IP precedence values and treated appropriately, depending on the WRED or CBWFQ drop probability configuration.

LSRs Used at the Core of an MPLS Network

Label switching routers (LSRs) used at the core of a Multiprotocol Label Switching (MPLS) network are devices running MPLS software. These devices at the core of an MPLS network process packets as follows:

- 1 MPLS labeled packets coming from the edge devices or other core devices enter the core device.
- 2 A lookup is done at the core device to determine the next hop LSR.
- 3 An appropriate label is placed (swapped) on the packet and the MPLS EXP bits are copied.
- 4 The labeled packet is then forwarded to the output interface for processing.
- 5 The packets are differentiated by the MPLS EXP field marking and treated appropriately, depending on the weighted early random detection (WRED) and class-based weighted fair queuing (CBWFQ) configuration.

Benefits of MPLS CoS in IP Backbones

You realize the following benefits when you use MPLS CoS in a backbone consisting of IP devices running Multiprotocol Label Switching (MPLS):

- Efficient resource allocation—Weighted fair queuing (WFQ) is used to allocate bandwidth on a per-class and per-link basis, thereby guaranteeing a percentage of link bandwidth for network traffic.
- Packet differentiation—When IP packets traverse an MPLS network, packets are differentiated by mapping the IP precedence bits of the IP packets to the MPLS CoS bits in the MPLS EXP field. This mapping of bits enables the service provider to maintain end-to-end network guarantees and meet the provisions of customer service level agreements (SLAs).

- Future service enhancements—MPLS CoS provides building blocks for future service enhancements (such as virtual leased lines) by meeting bandwidth requirements.

How to Configure MPLS Quality of Service

Configuring WRED

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **random-detect**
5. **random-detect precedence** *min-threshold max-threshold mark-probability*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>type number</i> Example: Device(config)# gigabitethernet0/0/0	Specifies the interface type and number, and enters interface configuration mode.
Step 4	random-detect Example: Device(config-if)# random-detect	Configures the interface to use weighted random early detection/distributed weighted random early detection (WRED/DWRED).
Step 5	random-detect precedence <i>min-threshold max-threshold mark-probability</i> Example: Device(config-if)# random-detect precedence 0 32 256 100	Configures WRED/DWRED parameters per precedence value.

	Command or Action	Purpose
Step 6	end Example: Device(config-if)# end	Returns to privileged EXEC mode.

Verifying WRED

To verify weighted random early detection (WRED), use a command of the form shown in the following table. This example is based on "Device2" in the network topology shown in the figure in the configuration examples section.

SUMMARY STEPS

1. **show queueing interface *subinterface***

DETAILED STEPS

show queueing interface *subinterface*

Example:

```
Device2# show queueing interface gigabitethernet6/0/0
```

Verifies the WRED configuration on the specified interface.

```
Device2# show queueing interface gigabitethernet6/0/0
```

```
Interface Gige6/0/0 queueing strategy:random early detection (WRED)
Exp-weight-constant:9 (1/512)
Mean queue depth:0
```

Class	Random drop	Tail drop	Minimum threshold	Maximum threshold	Mark probability
0	85	0	20	40	1/10
1	22	0	22	40	1/10
2	0	0	24	40	1/10
3	0	0	26	40	1/10
4	0	0	28	40	1/10
5	0	0	31	40	1/10
6	0	0	33	40	1/10
7	0	0	35	40	1/10
rsvp	0	0	37	40	1/10

Configuring CAR

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *name*
4. **rate-limit input** [**access-group** [**rate-limit**] *acl-index*] *bps burst-normal burst-max conform-action conform-action exceed-action exceed-action*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable Example: Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	interface <i>name</i> Example: Device(config)# interface gigabitethernet	Designates the input interface, and enters interface configuration mode.
Step 4	rate-limit input [access-group [rate-limit] <i>acl-index</i>] <i>bps burst-normal burst-max conform-action conform-action exceed-action exceed-action</i> Example: Device(config-if)# rate-limit input access-group 101 496000 32000 64000 conform-action set-prec-transmit 4	Specifies the action to take on packets during label imposition.
Step 5	end Example: Device(config-if)# end	Returns to privileged EXEC mode.

Verifying the CAR Configuration

SUMMARY STEPS

1. **show interfaces *slot/port* rate-limit**

DETAILED STEPS

show interfaces *slot/port* rate-limit

Example:

Device2# show interfaces fe1/1/1 rate-limit

Verifies the CAR configuration, use a command of the following form.

Device2# **show interfaces fe1/1/1 rate-limit**

```
FastEthernet1/1/1
  Input
    matches:access-group 101
    params: 496000 bps, 32000 limit, 64000 extended limit
    conformed 2137 packets, 576990 bytes; action:set-prec-transmit 4
    exceeded 363 packets, 98010 bytes; action:set-prec-transmit 0
    last packet:11788ms ago, current burst:39056 bytes
    last cleared 00:01:18 ago, conformed 58000 bps, exceeded 10000 bps
```

Configuring CBWFQ

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map *class-map-name***
4. **match *type number***
5. **policy-map *policy-map-name***
6. **class *class-map-name***
7. **bandwidth *number***
8. **interface *type number***
9. **service-policy output *policy-map-name***
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	enable	Enables privileged EXEC mode.

	Command or Action	Purpose
	Example: Device> enable	<ul style="list-style-type: none"> • Enter your password if prompted.
Step 2	configure terminal Example: Device# configure terminal	Enters global configuration mode.
Step 3	class-map <i>class-map-name</i> Example: Device(config)# class-map class-map-1	Creates a class map, and enters class-map configuration mode.
Step 4	match <i>type number</i> Example: Device(config-cmap)# match ip precedence 0 1	Specifies the traffic on which the class map is to match.
Step 5	policy-map <i>policy-map-name</i> Example: Device(config-cmap)# policy-map outputmap	Creates a policy map, and enters policy-map configuration mode.
Step 6	class <i>class-map-name</i> Example: Device(config-pmap)# class class-map-1	Associates the class map with the policy map.
Step 7	bandwidth <i>number</i> Example: Device(config-pmap-c)# bandwidth 10000	Associates the bandwidth (CBWFQ) action to act on traffic matched by the class map, and enters policy-map class configuration mode.
Step 8	interface <i>type number</i> Example: Device(config-pmap-c)# interface gigabitethernet0/0/0	Specifies the interface type and number, and enters interface configuration mode.
Step 9	service-policy output <i>policy-map-name</i> Example: Device(config-if)# service-policy output outputmap	Assigns the policy map to an interface.
Step 10	end Example: Device(config-if)# end	Returns to privileged EXEC mode.

Verifying the CBWFQ Configuration

SUMMARY STEPS

1. `show policy-map interface type number`

DETAILED STEPS

`show policy-map interface type number`

Example:

Device5# `show policy-map interface fe5/1/0`

Verifies the class-based weighted fair queueing (CBWFQ) configuration, use a command of the following form. This example is based on "Device 5" in the network topology shown in the figure in the configuration examples section.

Device5# `show policy-map interface fe5/1/0`

```
FastEthernet5/1/0
 service-policy output:outputmap
  class-map:prec_01 (match-all)
    522 packets, 322836 bytes
    5 minute rate 1000 bps
    match:ip precedence 0 1
    queue size 0, queue limit 1356
    packet output 522, packet drop 0
    tail/random drop 0, no buffer drop 0, other drop 0
    bandwidth:class-based wfq, weight 10
    random-detect:
      Exp-weight-constant:9 (1/512)
      Mean queue depth:0
  Class Random      Tail      Minimum      Maximum      Mark      Output
      drop      drop threshold threshold probability packets
  0          0          0          3390         6780         1/10         522
  1          0          0          3813         6780         1/10          0
  2          0          0          4236         6780         1/10          0
  3          0          0          4659         6780         1/10          0
  4          0          0          5082         6780         1/10          0
  5          0          0          5505         6780         1/10          0
  6          0          0          5928         6780         1/10          0
  7          0          0          6351         6780         1/10          0

  class-map:prec_23 (match-all)
    0 packets, 0 bytes
    5 minute rate 0 bps
    match:ip precedence 2 3
    queue size 0, queue limit 0
    packet output 0, packet drop 0
    tail/random drop 0, no buffer drop 0, other drop 0
    bandwidth:class-based wfq, weight 15
    random-detect:
      Exp-weight-constant:9 (1/512)
      Mean queue depth:0
  Class Random      Tail      Minimum      Maximum      Mark      Output
      drop      drop threshold threshold probability packets
  0          0          0          0            0            1/10         0
  1          0          0          0            0            1/10         0
  2          0          0          0            0            1/10         0
  3          0          0          0            0            1/10         0
  4          0          0          0            0            1/10         0
  5          0          0          0            0            1/10         0
  6          0          0          0            0            1/10         0
  7          0          0          0            0            1/10         0
```

```

class-map:prec_45 (match-all)
  2137 packets, 576990 bytes
  5 minute rate 16000 bps
  match:ip precedence 4 5
  queue size 0, queue limit 2712
  packet output 2137, packet drop 0
  tail/random drop 0, no buffer drop 0, other drop 0
  bandwidth:class-based wfq, weight 20
  random-detect:
    Exp-weight-constant:9 (1/512)
    Mean queue depth:0

```

Class	Random drop	Tail drop	Minimum threshold	Maximum threshold	Mark probability	Output packets
0	0	0	3390	6780	1/10	0
1	0	0	3813	6780	1/10	0
2	0	0	4236	6780	1/10	0
3	0	0	4659	6780	1/10	0
4	0	0	5082	6780	1/10	2137
5	0	0	5505	6780	1/10	0
6	0	0	5928	6780	1/10	0
7	0	0	6351	6780	1/10	0

```

class-map:prec_67 (match-all)
  0 packets, 0 bytes
  5 minute rate 0 bps
  match:ip precedence 6 7
  queue size 0, queue limit 0
  packet output 0, packet drop 0
  tail/random drop 0, no buffer drop 0, other drop 0
  bandwidth:class-based wfq, weight 25
  random-detect:
    Exp-weight-constant:9 (1/512)
    Mean queue depth:0

```

Class	Random drop	Tail drop	Minimum threshold	Maximum threshold	Mark probability	Output packets
0	0	0	0	0	1/10	0
1	0	0	0	0	1/10	0
2	0	0	0	0	1/10	0
3	0	0	0	0	1/10	0
4	0	0	0	0	1/10	0
5	0	0	0	0	1/10	0
6	0	0	0	0	1/10	0
7	0	0	0	0	1/10	0

```

class-map:class-default (match-any)
  0 packets, 0 bytes
  5 minute rate 0 bps
  match:any
    0 packets, 0 bytes
    5 minute rate 0 bps
  queue size 0, queue limit 4068
  packet output 90, packet drop 0
  tail/random drop 0, no buffer drop 0, other drop 0

```

Device5#
Device5# **show queuing interface fal/1/0**

```

Interface FastEthernet1/1/0 queuing strategy:VIP-based fair queuing
FastEthernet1/1/0 queue size 0
  pkts output 2756, wfq drops 0, nobuffer drops 0
WFQ:aggregate queue limit 13561 max available buffers 13561

Class 0:weight 30 limit 4068 qsize 0 pkts output 97 drops 0
Class 2:weight 10 limit 1356 qsize 0 pkts output 522 drops 0
Class 3:weight 15 limit 0 qsize 0 pkts output 0 drops 0
Class 4:weight 20 limit 2712 qsize 0 pkts output 2137 drops 0
Class 5:weight 25 limit 0 qsize 0 pkts output 0 drops 0 \

```

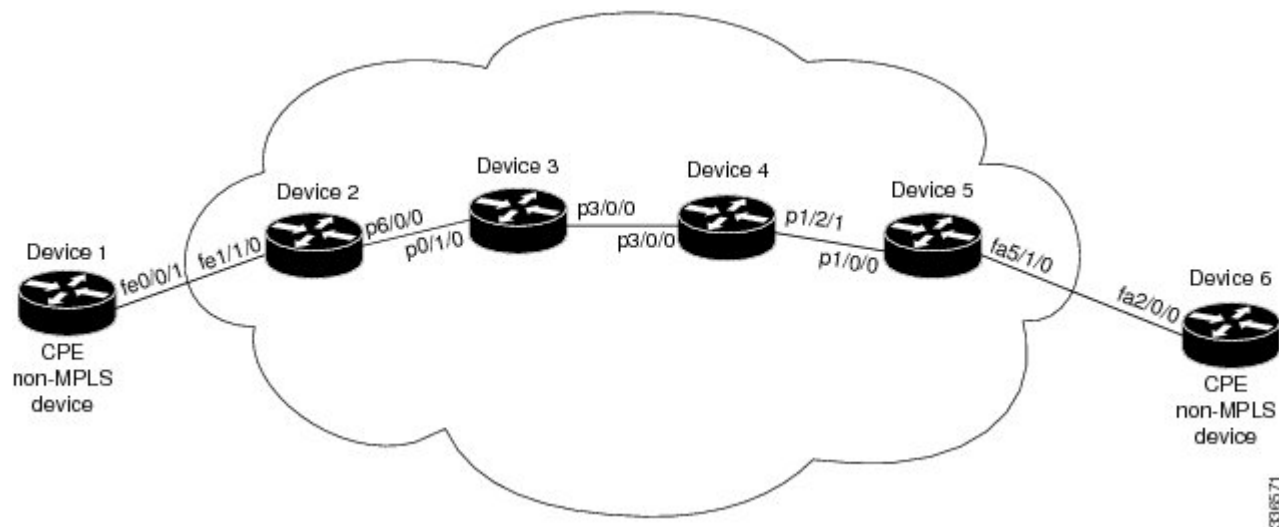
What to Do Next

.

Configuration Examples for MPLS Quality of Service

The configuration examples are based on the sample network topology shown in the figure below.

Figure 1: Sample Network Topology for Configuring MPLS CoS on Device Interfaces



Example: Configuring Cisco Express Forwarding

Cisco Express Forwarding must be running on all devices in the Multiprotocol Label Switching (MPLS) network for MPLS CoS to work. To enable Cisco Express Forwarding, use one of the following commands:

```
Device(config)# ip cef
OR
Device(config)# ip cef distributed
```

Example: Running IP on Device 1

The following commands enable IP routing on Device 1. All devices in the figure must have IP enabled. Device 1 is not part of the Multiprotocol Label Switching (MPLS) network.

```
!
ip routing
!
hostname R1
!
interface Loopback0
ip address 10.1.1.1 255.255.255.255
!
interface FastEthernet0/0/1
ip address 10.0.0.1 255.0.0.0
!
```

```
router ospf 100
 network 10.0.0.0 0.255.255.255 area 100
 network 10.0.0.1 0.255.255.255 area 100
```

Example: Running MPLS on Device 2

Device 2 is a label edge router. Cisco Express Forwarding and Multiprotocol Label Switching (MPLS) must be enabled on this device. Committed access rate (CAR) is also configured on Device 2 and Fast Ethernet interface 1/1/3. The CAR policy used at Fast Ethernet interface 1/1/0 acts on incoming traffic matching access-list 101. If the traffic rate is less than the committed information rate (in this example, 496000), the traffic will be sent with IP precedence 4. Otherwise, this traffic will be sent with IP precedence 0.

```
!
ip routing
!
hostname R2
!
ip cef
mpls ip
tag-switching advertise-tags
!
interface Loopback0
 ip address 10.10.10.10 255.255.255.255
!
interface FastEthernet1/1/0
 ip address 10.0.0.2 255.0.0.0
 rate-limit input access-group 101 496000 32000 64000 conform-action set-prec-transmit 4
 exceed-action set-prec-transmit 0
!
interface POS6/0/0
 ip address 10.0.0.1 255.0.0.0
 mpls label protocol ldp
 mpls ip
 random-detect
 clock source internal
!
router ospf 100
 network 10.0.0.0 0.255.255.255 area 100
 network 10.1.0.0 0.255.255.255 area 100
 network 11.0.1.0 0.255.255.255 area 100
!
access-list 101 permit ip host 10.10.1.1 any
```

Example: Running MPLS on Device 3

Device 3 is running Multiprotocol Label Switching (MPLS). Cisco Express Forwarding and MPLS must be enabled on this device.

```
!
ip routing
mpls ip
tag-switching advertise-tags
!
hostname R3
!
interface Loopback0
 ip address 10.10.10.10 255.255.255.255
!
interface POS0/1/0
 ip address 10.0.0.2 255.0.0.0
 mpls label protocol ldp
 mpls ip
 crc 16
!
interface POS3/0/0
```

Example: Running MPLS on Device 4

```

ip address 10.0.0.1 255.0.0.0
mpls label protocol ldp
mpls ip
crc 16
clock source internal
tx-cos stm16-rx
!
router ospf 100
network 10.0.1.0 0.255.255.255 area 100
network 10.0.0.1 0.255.255.255 area 100
network 10.1.0.0 0.255.255.255 area 100
!
cos-queue-group stm16-rx
precedence 0 random-detect-label 0
precedence 0 queue 0
precedence 1 queue 1
precedence 1 random-detect-label 1
precedence 2 queue 2
precedence 2 random-detect-label 2
precedence 3 random-detect-label 2
precedence 4 random-detect-label 2
precedence 5 random-detect-label 2
precedence 6 random-detect-label 2
precedence 7 queue low-latency
precedence 7 random-detect-label 2
random-detect-label 0 250 1000 1
random-detect-label 1 500 1250 1
random-detect-label 2 750 1500 1
queue 0 50
queue 1 100
queue 2 150
queue low-latency alternate-priority 500

```

Example: Running MPLS on Device 4

Device 4 is running Multiprotocol Label Switching (MPLS). Cisco Express Forwarding and MPLS must be enabled on this device.

```

!
ip routing
mpls ip
tag-switching advertise-tags
!
hostname R4
!
interface Loopback0
ip address 10.0.0.0 255.255.255.255
!
interface POS1/2/1
ip address 10.0.0.1 255.0.0.0
mpls label protocol ldp
mpls ip
crc 16
clock source internal
tx-cos stm16-rx
!
router ospf 100
network 10.0.0.0 0.255.255.255 area 100
network 10.1.0.0 0.255.255.255 area 100
network 10.0.1.0 0.255.255.255 area 100
!
cos-queue-group stm16-rx
precedence 0 queue 0
precedence 0 random-detect-label 0
precedence 1 queue 1
precedence 1 random-detect-label 1
precedence 2 queue 2
precedence 2 random-detect-label 2
precedence 3 random-detect-label 2
precedence 4 random-detect-label 2

```



```

precedence 5 random-detect-label 2
precedence 6 random-detect-label 2
precedence 7 queue low-latency
random-detect-label 0 250 1000 1
random-detect-label 1 500 1250 1
random-detect-label 2 750 1500 1
queue 0 50
queue 1 100
queue 2 150
queue low-latency alternate-priority 200

```

Example: Running MPLS on Device 5

Device 5 is running Multiprotocol Label Switching (MPLS). Cisco Express Forwarding and MPLS must be enabled on this device. Device 5 has class-based weighted fair queueing (CBWFQ) enabled on Fast Ethernet interface 5/1/0. In this example, class maps are created, matching packets with various IP precedence values. These class maps are then used in a policy map named “outputmap,” where CBWFQ is assigned to each class. Finally, the policy map is assigned to the outbound Fast Ethernet interface 5/1/0.

```

!
ip routing
mpls ip
tag-switching advertise-tags
!
hostname R5
!
!
class-map match-all prec_01
  match ip precedence 0 1
class-map match-all prec_23
  match ip precedence 2 3
class-map match-all prec_45
  match ip precedence 4 5
class-map match-all prec_67
  match ip precedence 6 7
!
!
policy-map outputmap
  class prec_01
    bandwidth 10000
    random-detect
  class prec_23
    bandwidth 15000
    random-detect
  class prec_45
    bandwidth 20000
    random-detect
  class prec_67
    bandwidth 25000
    random-detect
!
ip cef distributed
!
interface Loopback0
  ip address 10.0.0.0 255.255.255.255
  no ip directed-broadcast
!
interface POS1/1/0
  ip address 10.0.0.2 255.0.0.0
  ip route-cache distributed
  mpls label protocol ldp
  mpls ip
!
interface FastEthernet5/1/0
  ip address 10.0.0.1 255.0.0.0
  ip route-cache distributed
  full-duplex
  service-policy output outputmap

```

```

!
router ospf 100
 network 10.1.0.0 0.255.255.255 area 100
 network 10.0.1.0 0.255.255.255 area 100
 network 10.0.0.1 0.255.255.255 area 100

```

Example: Running IP on Device 6

Device 6 is running IP. Cisco Express Forwarding must be enabled on this device. Device 6 is not part of the Multiprotocol Label Switching (MPLS) network.

```

!
ip routing
!
hostname R6
!
ip cef distributed
!
interface Loopback0
 ip address 10.0.0.0 255.255.255.255
!
interface FastEthernet2/0/0
 ip address 10.0.0.2 255.0.0.0
 ip route-cache distributed
 full-duplex
!
router ospf 100
 network 10.0.0.0 0.255.255.255 area 100
 network 10.1.0.0 0.255.255.255 area 100
!

```

Example: Configuring WRED on a POS Interface for Cisco 12000 Series GSR Routers

In this example, weighted random early detection (WRED) is configured on a POS interface. The CoS queue group called “stm16-rx” is created, and settings are made to determine how closely the weighted average follows the instantaneous queue depth. The CoS queue group is applied to the transmit (tx) and receive (rx) directions. In the receive direction, a table indicating which cos-queue-group parameter sets to use for a given destination slot is created, and then the table is linked to the specified slot on which WRED is enabled.

```

Device(config)# cos-queue-group stm16-rx
Device(config-cos-que)# random-detect-label 0 250 1000 1
Device(config-cos-que)# random-detect-label 1 500 1250 1
Device(config-cos-que)# random-detect-label 2 750 1500 1
Device(config-cos-que)# precedence 0 random-detect-label 0
Device(config-cos-que)# precedence 1 random-detect-label 1
Device(config-cos-que)# precedence 2 random-detect-label 2
Device(config-cos-que)# precedence 3 random-detect-label 2
Device(config-cos-que)# precedence 4 random-detect-label 2
Device(config-cos-que)# precedence 5 random-detect-label 2
Device(config-cos-que)# precedence 6 random-detect-label 2
Device(config-cos-que)# precedence 7 random-detect-label 2
Device(config-cos-que)# exponential-weighting-constant 9
Device(config-if)# tx-cos stm16-tx
Device(config)# slot-table-cos stm16-rx-table
Device(config-slot-cos)# destination-slot all stm16-rx
Device (config-slot-cos)# exit
Device(config)# rx-cos-slot 1 stm16-rx-table

```

Example: Configuring MDRR on a POS Interface for Cisco 12000 Series GSR Routers

In this example, an MDRR cos-queue-group is created that maps IP precedences to MDRR queues and maps precedence 7 to a low-latency queue. Queue 0 has a weight value of 50, queue 2 has a weight value of 100, and queue 2 has a weight value of 150. The low-latency queue works in alternate-priority mode.

```
Device(config)# cos-queue-group stml6-rx
Device(config-cos-que)# precedence 0 queue 0
Device(config-cos-que)# precedence 1 queue 1
Device(config-cos-que)# precedence 2 queue 2
Device(config-cos-que)# precedence 7 queue low-latency
Device(config-cos-que)# queue 0 50
Device(config-cos-que)# queue 1 100
Device(config-cos-que)# queue 2 150
Device(config-cos-que)# queue low-latency alternate-priority 200
Device(config-cos-que)# exit
Device(config)#
```

Example: Configuring WRED and MDRR for Cisco 12000 Series GSR Routers

```
cos-queue-group stml6-rx
  random-detect-label 0 250 1000 1
  random-detect-label 1 500 1250 1
  random-detect-label 2 750 1500 1
  precedence 0 random-detect-label 0
  precedence 1 random-detect-label 1
  precedence 2 random-detect-label 2
  precedence 3 random-detect-label 2
  precedence 4 random-detect-label 2
  precedence 5 random-detect-label 2
  precedence 6 random-detect-label 2
  exponential-weighting-constant 9
  precedence 0 queue 0
  precedence 1 queue 1
  precedence 2 queue 2
  precedence 3 queue 1
  precedence 4 queue 1
  precedence 5 queue 1
  precedence 6 queue 2
  precedence 7 queue low-latency
  queue 0 50
  queue 1 100
  queue 2 150
  queue low-latency alternate-priority 200
exit
```

Additional References for MPLS Quality of Service

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases

Related Topic	Document Title
MPLS QoS commands	Cisco IOS Quality of Service Solutions Command Reference Cisco IOS Multiprotocol Label Switching Command Reference

MIBs

MIB	MIBs Link
<ul style="list-style-type: none"> • CISCO-WRED-MIB • CISCO-CAR-MIB 	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/support

Feature Information for MPLS Quality of Service

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 5: Feature Information for MPLS Quality of Service

Feature Name	Releases	Feature Information
MPLS Quality of Service	12.0(5)T 12.0(11)T 12.0(22)S 12.2(17b)SXA 12.2(8)T Cisco IOS XE Release 2.1	<p>The MPLS Quality of Service feature (formerly named as the MPLS CoS feature) enables you to provide differentiated services across an MPLS network. To satisfy a wide range of networking requirements, you can specify the class of service applicable to each transmitted IP packet. Different classes of service can be established for IP packets by setting the IP precedence bit in the header of each packet</p> <p>No new or modified commands were introduced.</p>

