MPLS Quality of Service (QoS)

This document describes the use of the MPLS class of service (CoS) functionality in an MPLS network.

Note MPLS Class of Service is referred to as MPLS Quality of Service (QoS). This name reflects the growth of MPLS to encompass a wider meaning and highlight the path towards future enhancements.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "Feature Information for MPLS Quality of Service" section on page 18.

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

Contents

Prerequisites for MPLS QoS

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.

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

- QoS features:
  - Weighted fair queueing (WFQ)—WFQ, 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, a congestion avoidance mechanism, 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.
  - 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 QoS

MPLS QoS 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

Table 1 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 queueing (WFQ). Packet classes are differentiated based on bandwidth requirements and finite delay characteristics.	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.

MPLS CoS enables you to duplicate Cisco IOS XE 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/MPLS Terminology

Table 2 lists the existing legacy tag switching terms and the new, equivalent 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)

MPLS CoS: LSRs Used at the Edge of an MPLS Network

LSRs used at the edge of an MPLS network backbone are routers running MPLS software. The edge LSRs can be at the ingress or the egress of the network.

At the ingress of an MPLS network, routers 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 (CLI) (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 router 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 queueing (CBWFQ)

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

At the egress of an MPLS network, routers 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 router 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.

MPLS CoS: LSRs Used at the Core of an MPLS Network

LSRs used at the core of an MPLS network are routers running MPLS software. These routers at the core of an MPLS network process packets as follows:

1. MPLS labeled packets coming from the edge routers or other core routers enter the core router.
2. A lookup is done at the core router 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 WRED and 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 routers running MPLS:

- Efficient resource allocation—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 QoS

This section contains the procedures listed below. All the procedures are listed as optional. However, once you decide to configure a specific MPLS QoS feature (for example, WRED) that procedure becomes required.

- Configuring WRED, page 5 (optional)
- Verifying WRED, page 6 (optional)
- Configuring CAR, page 7 (optional)
- Verifying the CAR Configuration, page 7 (optional)
- Configuring CBWFQ, page 8 (optional)
- Verifying the CBWFQ Configuration, page 9 (optional)

Configuring WRED

To configure weighted random early detection (WRED), use the commands shown in the following table.

DETAILED STEPS

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

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 “Router2” in the network topology shown in Figure 1.

Step 1 Verifies the WRED configuration on the specified interface.

show queueing interface *subinterface*

```
Router2# 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       Tail    Minimum    Maximum     Mark
                drop       drop  threshold  threshold  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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**`enable`** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**`configure terminal`** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**`interface`** *`name`* **Example:** `Router(config)# interface gigabitethernet`	Designates the input interface.
Step 4	**`rate-limit input`** [**`access-group`** [**`rate-limit`**]*`acl-index`*] *`bps burst-normal burst-max`* **`conform-action`** *`conform-action`* **`exceed-action`** *`exceed-action`* **Example:** `Router(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:** `Router(config-if)# end`	Exits interface configuration mode.

Verifying the CAR Configuration

To verify weighted random early detection (WRED), use a command of the form shown in the following table. This example is based on “Router2” in the network topology shown in Figure 1.

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

To verify the CAR configuration, use a command of the following form. This example is based on “Router 2” in the network topology shown in Figure 1.

```
Router2# 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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**class-map** *class-map-name* **Example:** `Router(config)# class-map class-map-1`	Creates a class-map.
Step 4	**match** *type number* **Example:** `Router(config-cmap)# match ip precedence 0 1`	Specifies the traffic on which the class-map is to match.
Step 5	**policy-map** *policy-map-name* **Example:** `Router(config-cmap)# policy-map outputmap`	Creates a policy map.
Step 6	**class** *class-map-name* **Example:** `Router(config-pmap)# class class-map-1`	Associates class-map with policy-map.
Step 7	**bandwidth** *number* **Example:** `Router(config-pmap-c)# bandwidth 10000`	Associates bandwidth (CBWFQ) action to act on traffic matched by class-map.

	Command or Action	Purpose
Step 8	**interface** *type number* **Example:** `Router(config-pmap-c)# interface gigabitethernet0/0/0`	Specifies the interface type and number.
Step 9	**service-policy output** *policy-map-name* **Example:** `Router(config-if)# service-policy output outputmap`	Assigns policy-map to interface.

Verifying the CBWFQ Configuration

Step 1 **show policy-map interface** *type/number*

To verify the CBWFQ configuration, use a command of the following form. This example is based on "Router 5" in the network topology shown in Figure 1.

```
Router5# 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
```

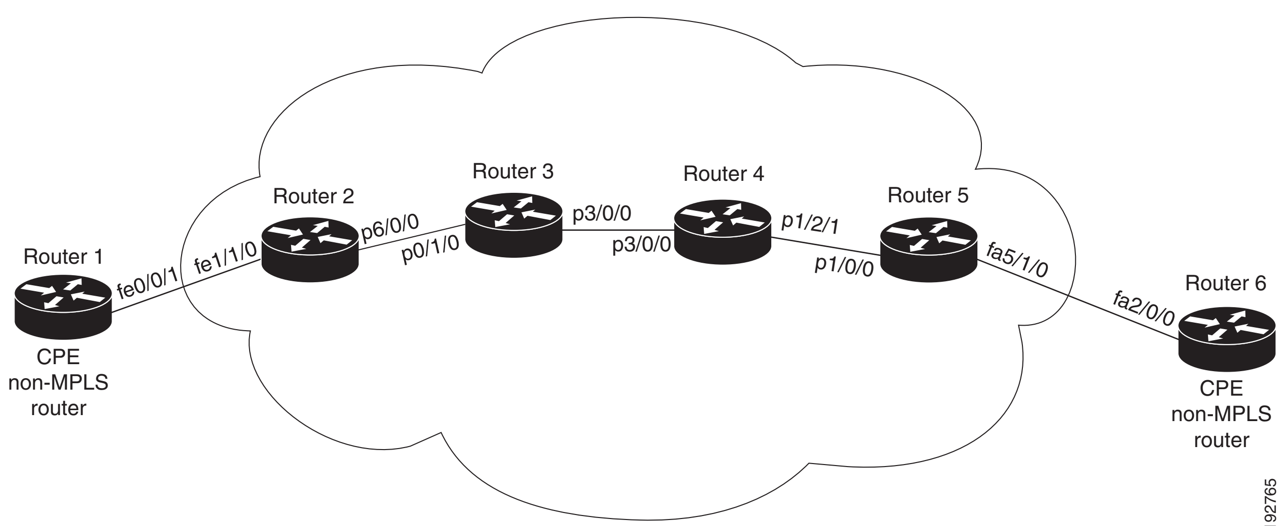
```
Class Random       Tail    Minimum    Maximum     Mark      Output
        drop       drop  threshold  threshold  probability packets
0          0          0          0          0    1/10            0
1          0          0          0          0    1/10            0
2          0          0          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       Tail    Minimum    Maximum     Mark      Output
        drop       drop  threshold  threshold  probability 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       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: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
```

```
Router5#
Router5# show queueing interface fa1/1/0

Interface FastEthernet1/1/0 queueing strategy:VIP-based fair queueing
  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 \
```

Configuration Examples for MPLS QoS

The configuration examples in this section are based on the sample network topology shown in Figure 1.

Figure 1 Sample Network Topology for Configuring MPLS CoS on Router Interfaces



Example: Configuring Cisco Express Forwarding

Cisco Express Forwarding is a prerequisite for using MPLS CoS; CEF must be running on all routers and switches in the MPLS network. To enable Cisco Express Forwarding on a router or a switch, use the following command:

```
Router(config)# ip cef
```

Example: Running IP on Router 1

The following commands enable IP routing on Router 1 (see Figure 1). All routers in Figure 1 must have IP enabled.

Note Router 1 is not part of the 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 Router 2

Router 2 (see Figure 1) is a label edge router. Cisco Express Forwarding and MPLS must be enabled on this router. CAR is also configured on Router 2 and interface e1/3. The CAR policy used at FastEthernet 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 Router 3

Router 3 (see Figure 1) is running MPLS. Cisco Express Forwarding and MPLS must be enabled on this router.

```
!
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
 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 Router 4

Router 4 (see Figure 1) is running MPLS. Cisco Express Forwarding and MPLS must be enabled on this router.

```
!
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 Router 5

Router 5 (see Figure 1) is running MPLS. Cisco Express Forwarding and MPLS must be enabled on this router. Router 5 has CBWFQ enabled on FastEthernet 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 FastEthernet 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 Router 6

Router 6 (see Figure 1) is running IP. Cisco Express Forwarding must be enabled on this router.

Note Router 6 is not part of the 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
!
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
MPLS QoS commands	• *Cisco IOS Quality of Service Solutions Command Reference* • *Cisco IOS Multiprotocol Label Switching Command Reference*
Quality of Service	*Quality of Service Overview*

Standards

Standard	Title
No new or modified standards are supported by the MPLS CoS feature.	—

MIBs

MIB	MIBs Link
• 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

RFCs

RFC	Title
No new or modified RFCs are supported by the MPLS CoS feature.	—

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/cisco/web/support/index.html

Feature Information for MPLS Quality of Service

Table 3 lists the release history for this feature lists and provides links to specific configuration information.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 2 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.

Table 3 Feature Information for MPLS Quality of Service

Feature Name	Releases	Feature Information
MPLS Class of Service (CoS)	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Aggregation Services Routers.

Glossary

ATM edge LSR—A router that is connected to the ATM-LSR cloud through LC-ATM interfaces. The ATM edge LSR adds labels to unlabeled packets and strips labels from labeled packets.

ATM-LSR—A label switch router with a number of LC-ATM interfaces. The router forwards the cells among these interfaces using labels carried in the VPI/VCI field.

CAR—Committed access rate (packet classification). CAR is the main feature supporting packet classification. CAR uses the type of service (ToS) bits in the IP header to classify packets. You can use the CAR classification commands to classify or reclassify a packet.

CoS—Class of service. A feature that provides scalable, differentiated types of service across an MPLS network.

IP precedence—A 3-bit value in a ToS byte used for assigning precedence to IP packets.

label—A short, fixed-length construct that tells switching nodes how to forward data (packets or cells).

label-controlled ATM interface (LC-ATM interface)—An interface on a router or switch that uses label distribution procedures to negotiate label VCs.

label imposition—The process of putting the first label on a packet.

label switch—A node that forwards units of data (packets or cells) on the basis of labels.

label-switched path (LSP)—An LSP results from a sequence of hops (Router 0...Router n) through which a packet travels from R0 to Rn by means of label switching mechanisms. A label-switched path can be determined dynamically (based on normal routing mechanisms), or it can be defined explicitly.

label-switched path (LSP) tunnel—A configured connection between two routers, in which label switching techniques are used for packet forwarding.

label switching router (LSR)—A Layer 3 router that forwards a packet based on the value of a label encapsulated in the packet.

label VC (LVC)—An ATM virtual circuit that is set up through ATM LSR label distribution procedures.

LBR—Label bit rate. A service category defined by this document for label-VC traffic. Link and per-VC bandwidth sharing can be controlled by relative bandwidth configuration at the edge and each switch along a label-VC. No ATM traffic-related parameters are specified.

LDP—Label Distribution Protocol. The protocol used to distribute label bindings to LSRs.

LFIB—Label forwarding information base. The data structure used by switching functions to switch labeled packets.

LIB—Label information base. A database used by an LSR to store labels learned from other LSRs, as well as labels assigned by the local LSR.

MPLS—Multiprotocol label switching. An emerging industry standard that defines support for MPLS forwarding of packets along normally routed paths (sometimes called MPLS hop-by-hop forwarding).

RED—Random early detection. A congestion avoidance algorithm in which a small percentage of packets are dropped when congestion is detected and before the queue in question overflows completely.

ToS bits—Type of service bits. A byte in the IPv4 header.

traffic engineering—The techniques and processes used to cause routed traffic to travel through the network on a path other than the one that would have been chosen if standard routing methods had been applied.

traffic engineering tunnel—A label-switched tunnel that is used for traffic engineering. Such a tunnel is set up through means other than normal Layer 3 routing; it is used to direct traffic over a path different from the one that Layer 3 routing would cause the tunnel to take.

VPN—Virtual private network. Enables IP traffic to use tunneling to transport data securely over a public TCP/IP network.

WRED—Weighted random early detection. A variant of RED in which the probability of a packet being dropped depends on either its IP precedence, CAR marking, or MPLS CoS (as well as other factors in the RED algorithm).

WFQ—Weighted fair queueing. A queue management algorithm that provides a certain fraction of link bandwidth to each of several queues, based on a relative bandwidth applied to each of the queues.