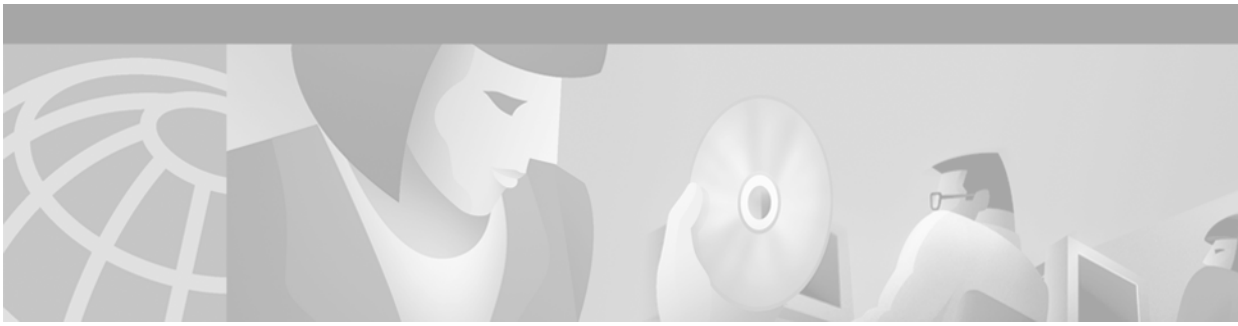


MPLS Class of Service (CoS)

This feature module describes the use of the MPLS class of service (CoS) functionality in an MPLS network. The document contains the following major sections:

- Feature Overview
- Supported Platforms
- Supported Standards, MIBs, and RFCs
- Prerequisites
- Configuration Tasks
- Configuration Examples for Cisco 7200/7500 Series Routers
- Configuration Examples for Cisco 12000 Series GSR Routers
- Glossary

The following table shows the change history of this document.

Release	Modification
12.0(5)T	This document was first introduced.
12.0(10)ST	MPLS CoS support has been added to this document for Cisco 12000 series GSR routers. Configuration tasks and examples for Cisco 7200 and 7500 series routers have been modified; configuration tasks and examples for Cisco12000 series GSR routers have been added.

Feature 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) for non-GSR platforms. 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.
	Modified deficit round robin (MDRR) for GSR platforms.	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.

For more information about configuring CoS functions (CAR, WRED, and WFQ), see the Cisco IOS *Quality of Service Solutions Configuration Guide*.

For complete command syntax information for CAR, WRED, and WFQ, see the Cisco IOS *Quality of Service Solutions Command Reference.*

MPLS CoS enables you to duplicate Cisco IOS 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

Table 2 Tag Switching Terms and Equivalent MPLS Terms (continued)

Old Designation	New Designation
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)

Supporting CoS in an MPLS Backbone

This section describes the following types of MPLS CoS configurations:

- LSRs Used at the Edge of an MPLS Network
- LSRs Used at the Core of an MPLS Network

LSRs Used at the Edge of an MPLS Network

LSRs used at the edge of an MPLS network backbone are usually Cisco 7200 or Cisco 7500 series 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 committed access rate (CAR) or some other classification mechanism, such as Modular QoS CLI (on the Cisco series 7200 and 7500 routers only), 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.

LSRs Used at the Core of an MPLS Network

LSRs used at the core of an MPLS network are usually Cisco GSR 12000 series routers or Cisco 7500 series 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.

Interfaces Supporting MPLS CoS Features

This section identifies the MPLS CoS features that are supported on various Cisco devices and interfaces.

Table 3 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[1]	Yes	N/A
Per-interface MDRR	N/A	N/A	Yes

1. This feature is supported in IOS release 12.1 and IOS release 12.1T

Table 4 lists the MPLS CoS features that are supported on ATM interfaces.

Table 4 MPLS CoS Features Supported on ATM Interfaces

MPLS CoS on ATM Cards	Cisco 7200 Series	Cisco 7500 Series	Cisco 12000 Series GSR
MPLS-WRED:			
Per interface	Yes[1]	Yes[1]	Yes
Per VC	No[2]	No[2]	No
MPLS-MDRR:			
Per interface	N/A	N/A	Yes
Per VC	N/A	N/A	No
MPLS-WFQ:			
Per interface, WFQ	Yes[1]	Yes[1]	No
Per interface, per-class WFQ	No	Yes	No

1. This feature is available on the ATM Lite port adapter (PA-A1).
2. This feature is available on the ATM Deluxe port adapter (PA-A3).

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.

Related Features and Technologies

You can use MPLS CoS functionality in any MPLS network.

Related Documents

For additional information about the following MPLS applications running on routers or switches in an MPLS network, consult the documentation listed below for Cisco IOS Release 12.0(10)ST and the *Quality of Service Solutions Configuration Guide* for Cisco IOS Release 12.0:

- *MPLS Label Distribution Protocol*—This feature enables peer label switching routers (LSRs) in an MPLS network to exchange label binding information for supporting hop-by-hop forwarding along normally routed paths.

- *Multiprotocol Label Switching on Cisco Routers*—This feature, implemented on Cisco routers and ATM switches, combines the performance of Layer 2 (data link layer) switching with the scalability of Layer 3 (network layer) routing. This combination enables service providers to handle the explosive growth now occurring in network utilization and to differentiate services without having to alter the existing network infrastructure. MPLS supports the dynamic creation of different routes between source and destination nodes, thus enabling IP services to be delivered efficiently by means of Internet backbones.
- *MPLS Traffic Engineering and Enhancements*—This feature enables an MPLS backbone to replicate and expand upon the traffic engineering capabilities of Layer 2 ATM and Frame Relay networks. In service provider and Internet service provider (ISP) backbones, traffic engineering provides an effective means of managing networks. Such backbones must support high transmission capacities and be resilient to link or node failures.
- *MPLS Egress Netflow Accounting*—This feature enables the detection of MPLS and IP flows traveling over links from the provider edge (PE) router to the customer edge (CE) router in a virtual private network (VPN).
- *AAL5 Transport Over MPLS*— This feature supports the transport of AAL5 protocol data units (PDUs) across an IP/MPLS backbone, providing an ATM permanent virtual circuit (PVC) transport service for PDUs with rate-limit policing and a configurable priority for PVCs.
- *MPLS Virtual Private Networks (VPNs)*—This feature enables users to deploy and administer IPv4 Layer 3, value-added services and business applications across a public network infrastructure. By deploying business applications on a broad scale over wide area networks (WANs), MPLS VPN users can reduce costs, increase revenue, and develop new business opportunities.
- *Quality of Service Solutions Configuration Guide* (Cisco IOS Release 12.0)—This document describes the quality of service (QoS) features in Cisco IOS and the service models that deliver QoS functionality. It also outlines the benefits that come from incorporating QoS functionality in your network and describes the Cisco IOS features that ensure better services to selected network traffic in Frame Relay, Asynchronous Transfer Mode (ATM), Ethernet, SONET, and IP-routed networks.

Supported Platforms

The routers that support MPLS CoS at the edge and core of a packet-based MPLS network are listed below:

- Cisco 7200 series routers
- Cisco 7500 series routers
- Cisco 12000 series GSR routers

Supported Standards, MIBs, and RFCs

Standards

No new or modified standards are supported by this MPLS CoS feature.

MIBs

- CISCO-WRED-MIB
- CISCO-CAR-MIB

For descriptions of supported MIBs and instructions for using MIBs, see the Cisco MIB website on CCO at http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml.

RFCs

No new or modified RFCs are supported by this MPLS CoS feature.

Prerequisites

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 (CEF)—CEF 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.

- Quality of Service (QoS) features:
  - Weighted fair queueing (WFQ)—WFQ, a dynamic scheduling method used on non-GSR platforms, 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.
  - Modified deficit round robin (MDRR)—MDRR, a traffic class prioritization mechanism used only on GSR platforms, 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: a) strict priority mode, or b) 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. CAR is supported on Cisco 7200 series routers, Cisco 7500 series routers, and Engine 0 cards on Cisco 12000 series GSR routers.

Configuration Tasks

This section describes the following configuration tasks:

- Configuring MPLS CoS on Cisco 7200/7500 Series Routers
  - Configuring WRED/DWRED on Cisco 7200/7500 Series Routers
  - Verifying WRED/DWRED on Cisco 7200/7500 Series Routers
  - Configuring CAR/DCAR on Cisco 7200/7500 Series Routers
  - Verifying CAR/DCAR Configuration on Cisco 7200/7500 Series Routers
  - Configuring CBWFQ on Cisco 7200/7500 Series Routers
  - Verifying CBWFQ Configuration on Cisco 7200/7500 Series Routers
- Configuring MPLS CoS on a Cisco 12000 Series GSR Router
  - Configuring WRED/MDRR on a Cisco 12000 Series GSR Router
  - Configuring Outgoing GSR Interfaces in the Transmit (frfab) Direction
  - Verifying WRED/MDRR in the Transmit (frfab) Direction
  - Configuring Incoming GSR Interfaces in the Receive (tofab) Direction
  - Verifying WRED/MDRR in the Receive (tofab) Direction

Configuring MPLS CoS on Cisco 7200/7500 Series Routers

Configuring WRED/DWRED on Cisco 7200/7500 Series Routers

To configure weighted random early detection (WRED) or distributed weighted random early detection (DWRED) on a Cisco 7200 or 7500 series router interface, issue the commands shown in the following table.

	Command	Purpose
Step 1	`Router(config)#` **`interface`** *`type number`*	Specifies the interface type and number.
Step 2	`Router(config-if)#` **`random-detect`**	Configures the interface to use WRED/DWRED.
Step 3	`Router(config-if)#` **`random-detect`** *`precedence min-threshold max-threshold mark-probability`*	Configures WRED/DWRED parameters per precedence value.

Verifying WRED/DWRED on Cisco 7200/7500 Series Routers

To verify weighted random early detection (WRED) or distributed weighted random early detection (DWRED) on a Cisco 7200 or 7500 series router interface, issue a command of the form shown in the following table. This example is based on "Router2" in the network topology shown in Figure 1.

Command — **Purpose**

Step 1 — Verifies WRED/MDRR configuration on the specified interface.

```
Router2# show queueing interface p6/0
Interface POS6/0 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/DCAR on Cisco 7200/7500 Series Routers

To configure committed access rate (CAR) or distributed committed access rate (DCAR) on a Cisco 7200 or 7500 series router interface, issue the commands shown in the following table.

	Command	Purpose
Step 1	`Router(config)#` **`interface`** *`name`*	Designates the input interface.
Step 2	`Router(config-int)#` **`rate-limit input`** `[`**`access-group`** `[`**`rate-limit`**`]`*`acl-index`*`]` *`bps burst-normal burst-max`* **`conform-action`** *`conform-action`* **`exceed-action`** *`exceed-action`*	Specifies the action to take on packets during label imposition.
Step 3	`Router(config-int)#` **`end`**	Exits interface configuration mode.

Verifying CAR/DCAR Configuration on Cisco 7200/7500 Series Routers

To verify CAR/DCAR configuration on a Cisco 7200 or 7500 series router interface, issue a command of the following form. This example is based on "Router 2" in the network topology shown in Figure 1.

```
Router2# show interfaces e1/3 rate-limit
Ethernet1/3
   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 on Cisco 7200/7500 Series Routers

To configure class-based weighted fair queueing (CBWFQ) on a Cisco 7200 or 7500 series router interface, issue the commands shown in the following table.

	Command	Purpose
Step 1	`Router(config)#` **`class-map`** *`class-map-name`*	Creates a class-map.
Step 2	`Router(config-cmap)#` **`match`** *`type number`*	Specifies the traffic on which the class-map is to match.
Step 3	`Router(config-cmap)#` **`policy-map`** *`policy-map-name`*	Creates a policy map.
Step 4	`Router(config-pmap)#` **`class`** *`class-map-name`*	Associates class-map with policy-map.
Step 5	`Router(config-pmap-c)#` **`bandwidth`** *`number`*	Associates bandwidth (CBWFQ) action to act on traffic matched by class-map.
Step 6	`Router(config-pmap-c)#` **`interface`** *`type number`*	Specifies the interface type and number.
Step 7	`Router(config-if)#` **`service-policy output`** *`policy-map-name`*	Assigns policy-map to interface.

Verifying CBWFQ Configuration on Cisco 7200/7500 Series Routers

To verify CBWFQ configuration on a Cisco 7200 or 7500 series router interface, issue 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 fa5/1/0
  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 fa5/1/0
Interface FastEthernet5/1/0 queueing strategy:VIP-based fair queueing
  FastEthernet5/1/0 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
```

Configuring MPLS CoS on a Cisco 12000 Series GSR Router

Configuring WRED/MDRR on a Cisco 12000 Series GSR Router

To configure WRED/MDRR on a Cisco 12000 series GSR router interface, issue the commands shown in the following table.

	Command	Purpose
Step 1	`Router(config)#` **`cos-queue-group`** *`cos-queue-group name`*	Creates a cos-queue-group. For an example of a complete cos-queue-group, see Step 1 in the table under the heading entitled "Configuring Outgoing GSR Interfaces in the Transmit (frfab) Direction."
Step 2	`Router(config)#` **`interface`** *`type number`*	Specifies the interface type and number.
Step 3	`Router(config-if)#` **`tx-cos`** *`cos-queue-group name`*	Assigns a cos-queue-group to the transmit interface.

Configuring Outgoing GSR Interfaces in the Transmit (frfab) Direction

To configure MPLS CoS on a Cisco 12000 series GSR router interface in the transmit direction, issue the commands shown in the following table.

	Command	Purpose

Step 1	`Router(config)# cos-queue-group cos-queue-group name`
	`Router(config-cos-que)# random-detect-label 0 25 100 1`
	`Router(config-cos-que)# random-detect-label 1 50 125 1`
	`Router(config-cos-que)# random-detect-label 2 75 150 1`
	`Router(config-cos-que)# precedence 0 random-detect-label 0`
	`Router(config-cos-que)# precedence 1 random-detect-label 1`
	`Router(config-cos-que)# precedence 2 random-detect-label 2`
	`Router(config-cos-que)# precedence 3 random-detect-label 2`
	`Router(config-cos-que)# precedence 4 random-detect-label 2`
	`Router(config-cos-que)# precedence 5 random-detect-label 2`
	`Router(config-cos-que)# precedence 6 random-detect-label 2`
	`Router(config-cos-que)# precedence 7 random-detect-label 2`
	`Router(config-cos-que)# exponential-weighting-constant 9`
		Creates a cos-queue-group.

Step 2	`Router(config)#` **`interface`** *`type number`*	Identifies the interface type on the router.
Step 3	`Router(config-if)#` **`ip address`** *`ip address`*	Assigns an IP address to the router interface.
Step 4	`Router(config-if)#` **`mpls ip`**	Enables MPLS on the router interface.
Step 5	`Router(config-if)#` **`mpls label protocol`** ***`ldp`***	Configures LDP (rather than TDP) as the label distribution protocol for the router interface.
Step 6	`Router(config)#` **`tx-cos`** *`cos-queue-group name`*	Assigns a cos-queue-group to the router interface.

Verifying WRED/MDRR in the Transmit (frfab) Direction

To verify WRED/MDRR configuration in the transmit direction on a Cisco 12000 series GSR router interface, issue a command of the following form.

```
Router# show interface pos6/0:1 random-detect
POS6/0:1
cos-queue-group: wred-test
RED Drop Counts
                    Tx Link                      To Fabric
RED Label     Random       Threshold          Random       Threshold
0              0              0                0              0
1              0              0                0              0
2              0              0                0              0
3          72913         345267                0              0
4              0              0                0              0
5         216225         433182                0              0
6              0              0                0              0
Tx-queue-limit drops: 0

Queue Lengths
Tx Queue (DRR configured) wred-test
Queue              Average             High Water Mark         Weight
0                    0.000                  0.000              10
1                    0.000                  0.000              10
2                    0.000                  0.000              10
3                  846.000                956.000              10
4                    0.000                  0.000              10
5                  873.000               1141.000              10
6                    0.000                  0.000              10
Low latency          0.000                  0.000              10

Tx RED config
  Precedence 0: not configured for drop
  Precedence 1: not configured for drop
  Precedence 2: not configured for drop
  Precedence 3:    500 min threshold,   1000 max threshold, 1/1 mark weight
  Precedence 4: not configured for drop
  Precedence 5:    700 min threshold,   1200 max threshold, 1/1 mark weight
  Precedence 6: not configured for drop
  Precedence 7: not configured for drop
weight 1/2
```

Configuring Incoming GSR Interfaces in the Receive (tofab) Direction

To configure MPLS CoS on a Cisco 12000 series GSR router interface in the receive direction, issue the commands shown in the following table.

	Command	Purpose
Step 1	`Router(config)#` **`slot-table-cos`** *`cos-queue-group name`*	Creates a slot table and links it to a cos-queue-group.
Step 2	`Router(config-slot-cos)#` **`destination slot`** *`slot number cos-queue-group name`*	Identifies the destination slots to which the specified cos-queue-group is to be applied.
Step 3	`Router(config-slot-cos)#` **`exit`**	Exits the slot-table-cos submode.
Step 4	`Router(config)#` **`rx-cos-slot`** *`line card number cos-queue-group name`*	Links the specified cos-queue-group to the specified router interface.

Verifying WRED/MDRR in the Receive (tofab) Direction

To verify the configuration of WRED/MDRR in the receive direction of an interface on a Cisco 12000 series GSR router interface, issue a command of the following form.

```
Router# show cos statistics (Supported on Engine 0 only)
Slot 4
---------------

Dest slot 6
cos-queue-group: wred-test
RED Drop Counts
                To Fabric
RED Label          Random       Threshold
0                  534885          250793
1                  538361          248846
2                  526379          259400
3                  520537          255598
4                       0               0
5                       0               0
6                       0               0

Queue Lengths

To Fabric Queues (DRR configured) wred-test
Queue              Average             High Water Mark         Weight
0                  712.000               5562.000              10
1                  702.000               7716.000              10
2                  702.000              11540.000              10
3                  753.000              14368.000              10
4                    0.000                  0.000              10
5                    0.000                  0.000              10
6                    0.000                  0.000              10
Low latency          0.000                  0.000              10

Tx RED config
  Precedence 0:    500 min threshold,   1000 max threshold, 1/1 mark weight
  Precedence 1:    500 min threshold,   1000 max threshold, 1/1 mark weight
  Precedence 2:    500 min threshold,   1000 max threshold, 1/1 mark weight
  Precedence 3:    500 min threshold,   1000 max threshold, 1/1 mark weight
  Precedence 4: not configured for drop
  Precedence 5: not configured for drop
  Precedence 6: not configured for drop
  Precedence 7: not configured for drop
weight 1/2

Slot 5
---------------

Dest slot 6
cos-queue-group: wred-test
RED Drop Counts
                To Fabric
RED Label          Random       Threshold
0                  554626          249651
1                  543842          263147
2                  538321          266974
3                  534728          258148
4                       0               0
5                       0               0
6                       0               0
```

```
Queue Lengths
To Fabric Queues (DRR configured) wred-test
Queue              Average             High Water Mark         Weight
0                  753.000               6539.000              10
1                  761.000              10190.000              10
2                  753.000              11272.000              10
3                  771.000               5980.000              10
4                    0.000                  0.000              10
5                    0.000                  0.000              10
6                    0.000                  0.000              10
Low latency          0.000                  0.000              10

Tx RED config
  Precedence 0:    500 min threshold,   1000 max threshold, 1/1 mark weight
  Precedence 1:    500 min threshold,   1000 max threshold, 1/1 mark weight
  Precedence 2:    500 min threshold,   1000 max threshold, 1/1 mark weight
  Precedence 3:    500 min threshold,   1000 max threshold, 1/1 mark weight
  Precedence 4: not configured for drop
  Precedence 5: not configured for drop
  Precedence 6: not configured for drop
  Precedence 7: not configured for drop
weight 1/2
```

Configuration Examples for Cisco 7200/7500 Series Routers

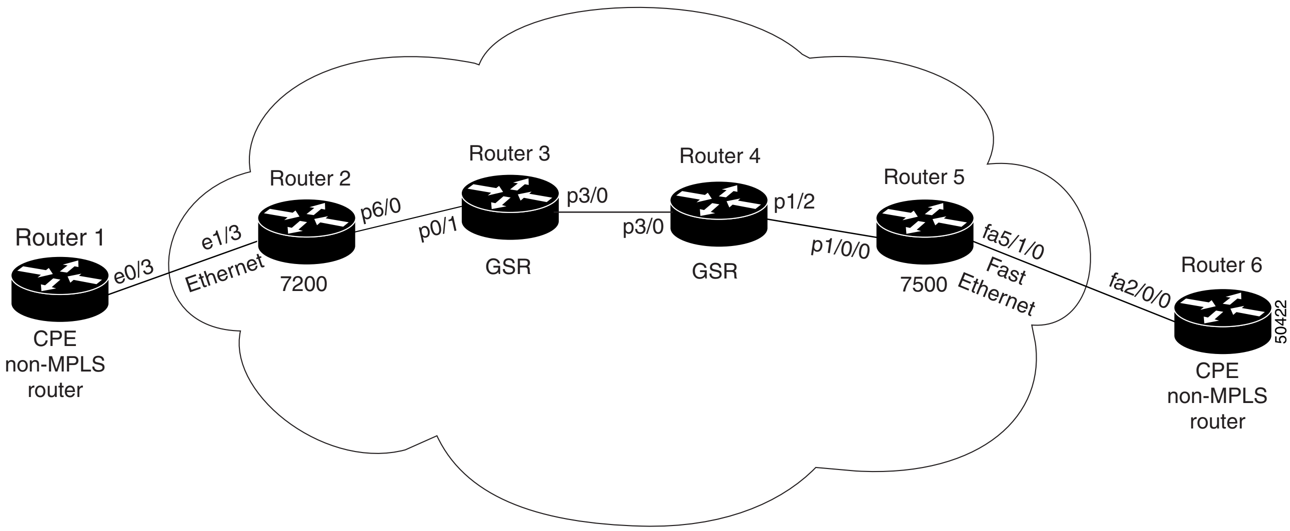
This section provides the following configuration examples for interfaces on Cisco 7200 and Cisco 7500 series routers.

- Configuring Cisco Express Forwarding
- Running IP on Router 1
- Running MPLS on Router 2
- Running MPLS on Router 3
- Running MPLS on Router 4
- Running MPLS on Router 5
- Running IP on Router 6

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

A later section entitled “Configuration Examples for Cisco 12000 Series GSR Routers” provides configuration examples for packet interfaces on Cisco 12000 series GSR routers. These configuration examples are likewise based on the sample network topology shown in Figure 1.

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



Configuring Cisco Express Forwarding

Cisco express forwarding (CEF) is a prerequisite for using MPLS CoS; CEF must be running on all routers and switches in the MPLS network. To enable CEF on a router or a switch, issue the following command, as appropriate:

`ip cef distributed` (for Cisco 7500 series routers)
!

`ip cef` (for Cisco 7200 series routers)
!

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 11.11.11.11 255.255.255.255
!
interface Ethernet0/3
 ip address 90.0.0.1 255.0.0.0
!
router ospf 100
 network 11.0.0.0 0.255.255.255 area 100
 network 90.0.0.0 0.255.255.255 area 100
```

Running MPLS on Router 2

Router 2 (see Figure 1) is a label edge router. CEF and MPLS must be enabled on this router. CAR is also configured on Router 2 and interface e1/3. The CAR policy used at interface e1/3 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 Ethernet1/3
 ip address 90.0.0.2 255.0.0.0
 rate-limit input access-group 101 496000 32000 64000 conform-action set-prec-transmit 4
exceed-action set-prec-transmit 0
!
interface POS6/0
 ip address 91.0.0.1 255.0.0.0
 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 90.0.0.0 0.255.255.255 area 100
 network 91.0.0.0 0.255.255.255 area 100
!
access-list 101 permit ip host 11.11.11.11 any
```

Running MPLS on Router 3

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

```
!
ip routing
mpls ip
tag-switching advertise-tags
!
hostname R3
!
interface Loopback0
 ip address 15.15.15.15 255.255.255.255
!
interface POS0/1
 ip address 91.0.0.2 255.0.0.0
 mpls label protocol ldp
 mpls ip
 crc 16
!
interface POS3/0
 ip address 92.0.0.1 255.0.0.0
 mpls label protocol ldp
 mpls ip
 crc 16
 clock source internal
 tx-cos stm16-rx
!
router ospf 100
 network 15.0.0.0 0.255.255.255 area 100
 network 91.0.0.0 0.255.255.255 area 100
 network 92.0.0.0 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
```

Running MPLS on Router 4

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

```
!
ip routing
mpls ip
tag-switching advertise-tags
!
hostname R4
!
interface Loopback0
 ip address 13.13.13.13 255.255.255.255
!
interface POS1/2
 ip address 93.0.0.1 255.0.0.0
 mpls label protocol ldp
 mpls ip
 crc 16
 clock source internal
 tx-cos stm16-rx
!
router ospf 100
 network 13.0.0.0 0.255.255.255 area 100
 network 92.0.0.0 0.255.255.255 area 100
 network 93.0.0.0 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
```

Running MPLS on Router 5

Router 5 (see Figure 1) is running MPLS. CEF and MPLS must be enabled on this router. Router 5 has CBWFQ enabled on interface fa5/1/0. 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 interface fa5/1/0.

```
!
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 12.12.12.12 255.255.255.255
 no ip directed-broadcast
!
interface POS1/1/0
 ip address 93.0.0.2 255.0.0.0
 ip route-cache distributed
 mpls label protocol ldp
 mpls ip
!
interface FastEthernet5/1/0
 ip address 94.0.0.1 255.0.0.0
 ip route-cache distributed
 full-duplex
 service-policy output outputmap
!
router ospf 100
 network 12.0.0.0 0.255.255.255 area 100
 network 93.0.0.0 0.255.255.255 area 100
 network 94.0.0.0 0.255.255.255 area 100
```

Running IP on Router 6

Router 6 (see Figure 1) is running IP. CEF 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 14.14.14.14 255.255.255.255
!
interface FastEthernet2/0/0
 ip address 94.0.0.2 255.0.0.0
 ip route-cache distributed
 full-duplex
!
router ospf 100
 network 14.0.0.0 0.255.255.255 area 100
 network 94.0.0.0 0.255.255.255 area 100
!
```

Configuration Examples for Cisco 12000 Series GSR Routers

This section provides the following configuration examples for packet interfaces on Cisco 12000 series GSR routers.

- Configuring WRED on a POS Interface
- Configuring MDRR on a POS Interface
- WRED/MDRR Configuration Example

These configuration examples are based on the sample MPLS network topology shown in Figure 1.

Configuring WRED on a POS Interface

To configure WRED on a POS interface, perform the following steps:

1. Create a WRED cos-queue-group:

```
Router(config)# cos-queue-group stm16-rx
(where stm16-rx is the cos-queue group to apply)
Router(config-cos-que)# random-detect-label 0 250 1000 1
Router(config-cos-que)# random-detect-label 1 500 1250 1
Router(config-cos-que)# random-detect-label 2 750 1500 1
Router(config-cos-que)# precedence 0 random-detect-label 0
Router(config-cos-que)# precedence 1 random-detect-label 1
Router(config-cos-que)# precedence 2 random-detect-label 2  Maps precedence
Router(config-cos-que)# precedence 3 random-detect-label 2  values to the set of
Router(config-cos-que)# precedence 4 random-detect-label 2  WRED parameters
Router(config-cos-que)# precedence 5 random-detect-label 2  to use.
Router(config-cos-que)# precedence 6 random-detect-label 2
Router(config-cos-que)# precedence 7 random-detect-label 2
Router(config-cos-que)# exponential-weighting-constant 9 Determines how closely weighted
                                                         average will follow instantaneous
                                                         queue depth.
```

2. Apply the cos-queue-group in transmit (TX) and receive (RX) directions.

   For the transmit (TX) direction, apply WRED parameters on the interface:

   ```
   Router(config-if)# tx-cos stm16-tx
   ```

   For the receive (RX) direction:

   a. Create a table indicating which cos-queue-group parameter sets to use for a given destination slot:

      ```
      Router(config)# slot-table-cos stm16-rx-table
      ```

      where *stm16-rx-table* is the label of the created table.

      ```
      Router(config-slot-cos)# destination-slot all stm16-rx
      Router (config-slot-cos)# exit
      ```

   b. Link the created table to the specified slot:

      ```
      Router(config)# rx-cos-slot 1 stm16-rx-table
      ```

      where *1* is the receive (RX) line on which WRED is enabled.

Configuring MDRR on a POS Interface

To configure MDRR on a POS interface, create an MDRR cos-queue-group, as shown in the following example:

```
Router(config)# cos-queue-group stm16-rx
Router(config-cos-que)# precedence 0 queue 0
Router(config-cos-que)# precedence 1 queue 1 (Maps IP precedences to MDRR queues)
Router(config-cos-que)# precedence 2 queue 2
Router(config-cos-que)# precedence 7 queue low-latency (Maps precedence 7 to low latency
queue)

Router(config-cos-que)# queue 0 50 (Queue 0 has weight value of 50)
Router(config-cos-que)# queue 1 100 (Queue 1 has weight value of 100)
Router(config-cos-que)# queue 2 150 (Queue 2 has weight value of 150)
Router(config-cos-que)# queue low-latency alternate-priority 200

(low-latency queue works in alternate-priority mode)
Router(config-cos-que)# exit
Router(config)#
```

WRED/MDRR Configuration Example

```
cos-queue-group stm16-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
```

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.