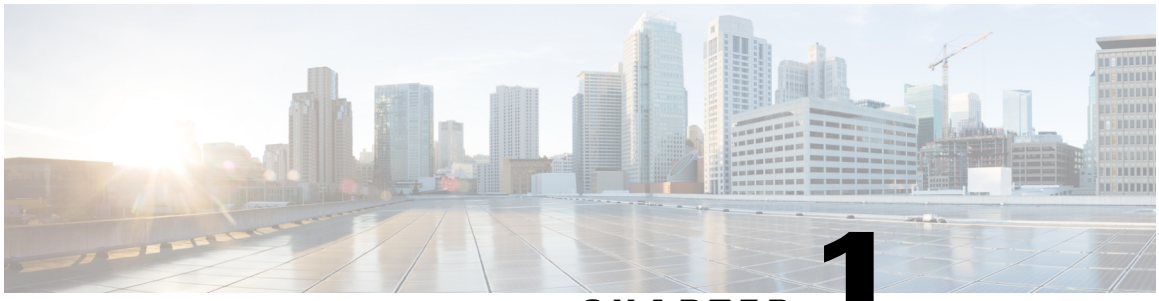
MPLS: Traffic Engineering: DiffServ Configuration Guide, Cisco IOS Release 15M&T

First Published: November 21, 2012

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS



CHAPTER 1

MPLS Traffic Engineering - DiffServ Aware (DS-TE)

The Multiprotocol Label Switching Traffic Engineering (MPLS TE) - DiffServ-Aware Traffic Engineering (DS-TE) feature enables service providers to perform separate admission control and separate route computation for discrete subsets of traffic (for example, voice and data traffic).

When DS-TE is combined with other Cisco software features such as QoS, the service provider can:

- Develop QoS services for end customers based on *signaled* rather than *provisioned* QoS
- Build the higher-revenue generating “strict-commitment” QoS services, without over-provisioning
- Offer virtual IP leased-line, Layer 2 service emulation, and point-to-point guaranteed bandwidth services including voice-trunking
- Enjoy the scalability properties offered by MPLS.

Information About MPLS Traffic Engineering - DiffServ Aware (DS-TE)

MPLS TE and Constraint-Based Routing (CBR)

MPLS TE allows constraint-based routing (CBR) of IP traffic. One of the constraints satisfied by CBR is the availability of required bandwidth over a selected path. DiffServ-aware TE extends MPLS traffic engineering to enable you to perform constraint-based routing of "guaranteed" traffic, which satisfies a more restrictive bandwidth constraint than that satisfied by CBR for regular traffic. The more restrictive bandwidth is termed a *sub-pool*, while the regular TE tunnel bandwidth is called the *global pool*. (The sub-pool is a portion of the global pool. In the new IETF-Standard, the global pool is called BC0 and the sub-pool is called BC1. These are two of an eventually available eight Class Types). This ability to satisfy a more restrictive bandwidth constraint translates into an ability to achieve higher Quality of Service (QoS) performance in terms of delay, jitter, or loss for the guaranteed bandwidth services end-to-end across the network.

DS-TE has been augmented to conform to IETF standards that were developed after the initial creation of Cisco DS-TE. Now both the traditional and the IETF versions of DS-TE can be run on your network; the new releases are backwards compatible.

For example, DS-TE can be used to ensure that traffic is routed over the network so that, on every link, there is never more than 40 per cent (or any assigned percentage) of the link capacity of guaranteed traffic (for example, voice), while there can be up to 100 per cent of the link capacity of regular traffic. Assuming that QoS mechanisms are also used on every link to queue guaranteed traffic separately from regular traffic, it then becomes possible to enforce separate "overbooking" ratios for guaranteed and regular traffic. In fact, for the guaranteed traffic it becomes possible to enforce no overbooking at all--or even an underbooking--so that very high QoS can be achieved end-to-end for that traffic, even while for the regular traffic a significant overbooking continues to be enforced.

Also, through the ability to enforce a maximum percentage of guaranteed traffic on any link, the network administrator can directly control the end-to-end QoS performance parameters without having to rely on over-engineering or on expected shortest path routing behavior. This is essential for transport of applications that have very high QoS requirements such as real-time voice, virtual IP leased line, and bandwidth trading, where over-engineering cannot be assumed everywhere in the network.

The new IETF-Standard functionality of DS-TE expands the means for allocating constrained bandwidth into two distinct models, called the "Russian Dolls Model" and the "Maximum Allocation Model". They differ from each other as follows:

Table 1: Bandwidth Constraint Model Capabilities

MODEL	Achieves Bandwidth Efficiency	Ensures Isolation across Class Types	Protects against QoS Degradation...		
		When Preemption is Not Used	When Preemption is Used	...of the Premium Class Type	...of all other Class Types
Maximum Allocation	Yes	Yes	Yes	Yes	**No**

MODEL	Achieves Bandwidth Efficiency	Ensures Isolation across Class Types	Protects against QoS Degradation...		
Russian Dolls	Yes	**No**	Yes	Yes	Yes

Therefore in practice, a Network Administrator might prefer to use:

- the Maximum Allocation Model when s/he needs to ensure isolation across all Class Types without having to use pre-emption, and s/he can afford to risk some QoS degradation of Class Types other than the Premium Class.
- the Russian Dolls Model when s/he needs to prevent QoS degradation of all Class Types and can impose pre-emption.

DS-TE involves extending OSPF (Open Shortest Path First routing protocol), so that the available sub-pool or class-type bandwidth at each preemption level is advertised in addition to the available global pool bandwidth at each preemption level. And DS-TE modifies constraint-based routing to take this more complex advertised information into account during path computation.

With the addition of IETF-Standard functionality (beginning with Cisco IOS Release 12.2(33)SRB), networks may accomplish DS-TE in three different combinations or "modes", so that they may transition to the IETF-Standard formats in a manner that will not degrade their ongoing traffic service. These three situations or modes are summarized as follows:

1. The original, or "Traditional" (pre-IETF-Standard) mode. This describes networks that already operate the form of DS-TE that was introduced by Cisco a few years ago. Such networks can continue to operate is this traditional mode, even when they use the new Release 12.2(33)SRB and subsequent releases.
2. The "Migration" or combination mode. Networks already running traditional DS-TE that would like to upgrade to the IETF-Standard should first configure their routers into the Migration mode. This will allow them to continue to operate DS-TE without tunnels being torn down. In Migration mode, routers will continue to generate IGP and tunnel signalling as in the Traditional form, but now these routers will add TE-class mapping and will accept advertisement in both the Traditional and the new IETF-Standard formats.
3. The "Liberal IETF" mode. Networks already running in the Migration mode can then move into IETF formats by reconfiguring their routers into this flexible (hence "Liberal") combination: their routers will henceforth generate IGP advertisement and tunnel signalling according to the new IETF Standard, but they will remain capable of accepting advertisement in the Traditional format, as well as in the new IETF format.

The table below summarizes these distinctions among the three modes.

Table 2: Summary of DS-TE Mode behaviors

	Uses TE-class mapping	Generates	Processes		
MODE		IGP Advertisement	RSVP-TE Signaling	IGP Advertisement	RSVP-TE Signaling
Traditional	No	traditional	traditional	traditional1	traditional

	Uses TE-class mapping	Generates	Processes		
Migration	Yes	traditional	traditional	traditional & IETF	traditional & IETF
Liberal IETF	Yes	IETF	traditional & IETF	traditional & IETF	traditional & IETF

1Note that it is not possible for the Traditional mode to be liberal in what it accepts in terms of IGP, since it does not use TE-Class mapping and therefore cannot interpret the “Unreserved Bandwidth” in the IETF-compliant way when the Subpool Sub-TLV is absent.

From Traditional to IETF-Standard Commands

DS-TE commands originally were developed from the then-existing command set that had been used to configure MPLS traffic engineering. The only difference introduced at that time to create DS-TE was the expansion of two commands:

- **ip rsvp bandwidth** was expanded to configure the size of the sub-pool on every link.
- **tunnel mpls traffic-eng bandwidth** was expanded to enable a TE tunnel to reserve bandwidth from the sub-pool.

The ip rsvp bandwidth command

The early MPLS command had been

```
ip rsvp bandwidth x y
```
where x = the size of the only possible pool, and y = the size of a single traffic flow (ignored by traffic engineering).

Then, to create the original implementation of DS-TE, the command was made into

```
ip rsvp bandwidth x y sub-pool z
```
where x = the size of the global pool, and z = the size of the sub-pool.

With the addition of the IETF-Standard version of DS-TE, the command has been further extended to become:

```
ip rsvp bandwidth x y [ [rdm x {subpool z | bc1 z}] | [mam bc0 x bc1 z]]
```
where x = the size of the global pool (now called **bc0**), and z = the size of the sub-pool (now called also **bc1**).

Two bandwidth constraint models also have become available, “Russian Dolls” (indicated by the keyword **rdm**) and “Maximum Allocation” (**mam**). The former model allows greater sharing of bandwidth across all Class Types (bandwidth pools), while the latter protects especially the premium Class Type. (The IETF Standard makes possible the future implementation of as many as seven sub-pools within one LSP, instead of just one sub-pool per LSP).

The tunnel mpls traffic-eng bandwidth command

The pre-DS-TE traffic engineering command was

```
tunnel mpls traffic-eng bandwidth b
```
where b = the amount of bandwidth this tunnel requires.

So for the original DS-TE, you specified from which pool (global or sub) the tunnel's bandwidth would come. You could enter

```
tunnel mpls traffic-eng bandwidth sub-pool b
```
to indicate that the tunnel should use bandwidth from the sub-pool. Alternatively, you could enter

```
tunnel mpls traffic-eng bandwidth b
```
to indicate that the tunnel should use bandwidth from the global pool (which was the default).

With the addition of the IETF-Standard version of DS-TE, the command has been extended to become:

```
tunnel mpls traffic-eng bandwidth [sub-pool|class-type 1] b
```
where both **sub-pool** and **class-type 1** indicate the same, smaller bandwidth pool (now called class-type 1). The two keywords can be used interchangeably.

The mpls traffic-eng ds-te commands

The IETF Standard introduces two new commands, one to indicate the Bandwidth Constraints model

```
mpls traffic-eng ds-te bc-model [rdm | mam]
```
and one to select the DS-TE mode:

```
mpls traffic-eng ds-te mode [migration|ietf]
```
(The concepts of bc-model and DS-TE mode were explained in the section above).

The first command allows you to select between the Russian Dolls Model (**rdm**) and the Maximum Allocation Model (**mam**) of bandwidth constraints.

The second command allows you to transition a network from traditional DS-TE tunnels to the IETF Standard without disrupting any of the tunnels' operation. To accomplish this, you first put the routers into Migration mode (using the **migration** keyword) and subsequently into the Liberal-IETF mode (using the **ietf** keyword).

Transitioning a Network to the IETF Standard

Networks already operating DS-TE tunnels by means of the traditional, pre-IETF-Standard software can switch to the IETF-Standard without interrupting their DS-TE service by following this sequence:

1. Install Cisco IOS Release 12.2(33)SRB (or a subsequent release) on each router in the network, gradually, one router at a time, using Cisco's In Service Software Upgrade (ISSU) procedure which protects ongoing network traffic from interruption. (After that installation, DS-TE tunnels in the network will continue to operate by using the pre-IETF-Standard formats.)
2. Enter the global configuration command **mpls traffic-eng ds-te mode migration** on each router in the network, one router at a time. This will enable the routers to receive IETF-format IGP advertisement and RSVP-TE signaling, while the routers will continue to generate and receive the pre-Standard formats for those two functions.
3. After all the routers in the network have begun to operate in Migration mode, enter the global configuration command **mpls traffic-eng ds-te mode ietf** on each router, one at a time. This will cause the router to refresh its TE tunnels with IETF-compliant Path signaling, without disrupting the tunnels' operation. This mode also causes the router to generate IGP advertisement in the IETF-Standard format.

Guaranteed Bandwidth Service Configuration

Once two bandwidth pools are configured traffic can be managed in the following ways:

- Use one pool, the sub-pool, for tunnels that carry traffic requiring strict bandwidth guarantees or delay guarantees
- Use the other pool, the global pool, for tunnels that carry traffic requiring only Differentiated Service.

Having a separate pool for traffic requiring strict guarantees allows you to limit the amount of such traffic admitted on any given link. Often, it is possible to achieve strict QoS guarantees only if the amount of guaranteed traffic is limited to a portion of the total link bandwidth.

Having a separate pool for other traffic (best-effort or diffserv traffic) allows you to have a separate limit for the amount of such traffic admitted on any given link. This is useful because it allows you to fill up links with best-effort/diffserv traffic, thereby achieving a greater utilization of those links.

Providing Strict QoS Guarantees Using DS-TE Sub-pool Tunnels

A tunnel using sub-pool bandwidth can satisfy the stricter requirements if you do all of the following:

- Select a queue--or in diffserv terminology, select a PHB (per-hop behavior)--to be used exclusively by the strict guarantee traffic. This shall be called the “GB queue.”
  If delay/jitter guarantees are sought, the diffserv Expedited Forwarding queue (EF PHB) is used. (On the Cisco 7500 [VIP], it is the "priority" queue.) You must configure the bandwidth of the queue to be at least equal to the bandwidth of the sub-pool.

  If only bandwidth guarantees are sought, the diffserv Assured Forwarding PHB (AF PHB) is used. (On the Cisco 7500 [VIP], you use one of the existing Class-Based Weighted Fair Queuing [CBWFQ] queues.)
- Ensure that the guaranteed traffic sent through the sub-pool tunnel is placed in the GB queue *at the outbound interface of every tunnel hop* , and that no other traffic is placed in this queue. This is done by marking the traffic that enters the tunnel with a unique value in the *mpls exp bits* field, and steering only traffic with that marking into the GB queue.
- Ensure that this GB queue is never oversubscribed; that is, see that no more traffic is sent into the sub-pool tunnel than the GB queue can handle.
  This done by rate-limiting the guaranteed traffic before it enters the sub-pool tunnel. The aggregate rate of all traffic entering the sub-pool tunnel should be less than or equal to the bandwidth capacity of the sub-pool tunnel. Excess traffic can be dropped (in the case of delay/jitter guarantees) or can be marked differently for preferential discard (in the case of bandwidth guarantees).
- Ensure that the amount of traffic entering the GB queue is limited to an appropriate percentage of the total bandwidth of the corresponding outbound link. The exact percentage to use depends on several factors that can contribute to accumulated delay in your network: your QoS performance objective, the total number of tunnel hops, the amount of link fan-in along the tunnel path, burstiness of the input traffic, and so on.
  This is done by setting the sub-pool bandwidth of each outbound link to the appropriate percentage of the total link bandwidth (that is, by adjusting the z parameter of the **ip rsvp bandwidth** command).

Providing Differentiated Service Using DS-TE Global Pool Tunnels

You can configure a tunnel using global pool bandwidth to carry best-effort as well as several other classes of traffic. Traffic from each class can receive differentiated service if you do all of the following:

1. Select a separate queue (a distinct diffserv PHB) for each traffic class. For example, if there are three classes (gold, silver, and bronze) there must be three queues (diffserv AF2, AF3, and AF4).
2. Mark each class of traffic using a unique value in the MPLS experimental bits field (for example gold = 4, silver = 5, bronze = 6).
3. Ensure that packets marked as Gold are placed in the gold queue, Silver in the silver queue, and so on. The tunnel bandwidth is set based on the expected aggregate traffic across all classes of service.

To control the amount of diffserv tunnel traffic you intend to support on a given link, adjust the size of the global pool on that link.

Providing Strict Guarantees and Differentiated Service in the Same Network

Because DS-TE allows simultaneous constraint-based routing of sub-pool and global pool tunnels, strict guarantees and diffserv can be supported simultaneously in a given network.

Prerequisites for MPLS Traffic Engineering - DiffServ Aware (DS-TE)

Your network must support the following Cisco software features in order to support guaranteed bandwidth services based on DiffServ-aware Traffic Engineering:

- MPLS
- IP Cisco Express Forwarding (CEF)
- OSPF or ISIS
- RSVP-TE
- QoS

How to Configure MPLS Traffic Engineering - DiffServ Aware (DS-TE)

Configuring DS-TE Tunnels

To establish a sub-pool (BC1) traffic engineering tunnel, you must enter configurations at three levels:

- the device level (router or switch router)

- the physical interface
- the tunnel interface

On the first two levels, you activate traffic engineering; on the third level--the tunnel interface--you establish the sub-pool tunnel. Therefore, it is only at the tunnel headend device that you need to configure all three levels. At the tunnel midpoints and tail, it is sufficient to configure the first two levels.

In the tables below, each command is explained in brief. For a more complete explanation of any command, type it into the Command Lookup Tool at http://www.cisco.com/cgi-bin/Support/Cmdlookup/home.pl . (If prompted to log in there, use your Cisco.com account username and password).

Level 1 Configuring the Device

At this level, you tell the device (router or switch router) to use accelerated packet-forwarding (known as Cisco Express Forwarding or CEF), MultiProtocol Label Switching (MPLS), traffic-engineering tunneling, a bandwidth constraints model, and either the OSPF or IS-IS routing algorithm (Open Shortest Path First or Intermediate System to Intermediate System). This level is called the global configuration mode, because the configuration is applied globally, to the entire device, rather than to a specific interface or routing instance.

You enter the following commands:

SUMMARY STEPS

1. Router(config)# **ip cef distributed**
2. Router(config)# **mpls traffic-eng tunnels**
3. Router(config)# **mpls traffic-eng ds-te bc-model** [**rdm** | **mam**]
4. Choose one of the following:
   - Router(config)# **router ospf**
   - Router(config)# **router isis**
5. Router (config-router)# **net** *network-entity-title*
6. Router (config-router)# **metric-style wide**
7. Router (config-router)# **is-type level** *n*
8. Router (config-router)# **mpls traffic-eng level***n*
9. Router (config-router)# **passive-interface loopback0**
10. Router(config-router)# **mpls traffic-eng router-id loopback0**
11. Router(config-router)# **mpls traffic-eng area** *num*

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# **ip cef distributed**	Enables CEF--which accelerates the flow of packets through the device.

	Command or Action	Purpose
Step 2	Router(config)# **mpls traffic-eng tunnels**	Enables MPLS, and specifically its traffic engineering tunnel capability.
Step 3	Router(config)# **mpls traffic-eng ds-te bc-model** [**rdm** | **mam**]	Specifies the bandwidth constraints model (see the Feature Overview section).
Step 4	Choose one of the following: • Router(config)# **router ospf** • Router(config)# **router isis**	Invokes the OSPF routing process for IP and puts the device into router configuration mode. Proceed now to Steps 10 and 11. Alternatively, you may invoke the IS-IS routing process with this command, and continue with Step 5.
Step 5	Router (config-router)# **net** *network-entity-title*	Specifies the IS-IS network entity title (NET) for the routing process.
Step 6	Router (config-router)# **metric-style wide**	Enables the router to generate and accept IS-IS new-style TLVs (type, length, and value objects).
Step 7	Router (config-router)# **is-type level** *n*	Configures the router to learn about destinations inside its own area or "IS-IS level".
Step 8	Router (config-router)# **mpls traffic-eng level1***n*	Specifies the IS-IS level (which must be same level as in the preceding step) to which the router will flood MPLS traffic-engineering link information.
Step 9	Router (config-router)# **passive-interface loopback0**	Instructs IS-IS to advertise the IP address of the loopback interface without actually running IS-IS on that interface. Continue with Step 10 but don't do Step 11--because Step 11 refers to OSPF.
Step 10	Router(config-router)# **mpls traffic-eng router-id loopback0**	Specifies that the traffic engineering router identifier is the IP address associated with the *loopback0* interface.
Step 11	Router(config-router)# **mpls traffic-eng area** *num*	Turns on MPLS traffic engineering for a particular OSPF area.

Level 2 Configuring the Physical Interface

Having configured the device, you now must configure the interface on that device through which the tunnel will run. To do that, you first put the router into interface-configuration mode.

You then enable Resource Reservation Protocol (RSVP). This protocol is used to signal (set up) a traffic engineering tunnel, and to tell devices along the tunnel path to reserve a specific amount of bandwidth for the traffic that will flow through that tunnel. It is with this command that you establish the maximum size of the sub-pool (BC1).

Finally, you enable the MPLS traffic engineering tunnel feature on this physical interface--and if you will be relying on the IS-IS routing protocol, you enable that as well .

To accomplish these tasks, you enter the following commands:

SUMMARY STEPS

1. Router(config)# **interface** *interface-id*
2. Router(config-if)# **ip rsvp bandwidth** [*interface-kbps*] [*single-flow-kbps*][[**rdm** *kbps*{[**subpool** *kbps*]|[**bc1** *subpool*]}]|[**mam max-reservable-bw** *kbps* **bc0** *kbps* **bc1** *kbps*]]
3. Router(config-if)# **mpls traffic-eng tunnels**
4. Router(config-if)# **ip router isis**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# **interface** *interface-id*	Moves configuration to the interface level, directing subsequent configuration commands to the specific interface identified by the *interface-id*.
Step 2	Router(config-if)# **ip rsvp bandwidth** [*interface-kbps*] [*single-flow-kbps*][[**rdm** *kbps*{[**subpool** *kbps*]|[**bc1** *subpool*]}]|[**mam max-reservable-bw** *kbps* **bc0** *kbps* **bc1** *kbps*]] **Example:**	Enables RSVP on this interface, indicates the Bandwidth Constraints Model to be used (explained in the Feature Overview section), and limits the amount of bandwidth RSVP can reserve on this interface. The sum of bandwidth used by all tunnels on this interface cannot exceed *interface-kbps.*
Step 3	Router(config-if)# **mpls traffic-eng tunnels**	Enables the MPLS traffic engineering tunnel feature on this interface.
Step 4	Router(config-if)# **ip router isis**	Enables the IS-IS routing protocol on this interface. Do not enter this command if you are configuring for OSPF.

Level 3 Configuring the Tunnel Interface

Now you create a set of attributes for the tunnel itself; those attributes are configured on the “tunnel interface” (not to be confused with the physical interface just configured above).

You enter the following commands:

SUMMARY STEPS

1. Router(config)# **interface tunnel1**
2. Router(config-if)# **tunnel destination** *A.B.C.D*
3. Router(config-if)# **tunnel mode mpls traffic-eng**
4. Router(config-if)# **tunnel mpls traffic-eng bandwidth** {**sub-pool** | **class-type1**} *bandwidth*
5. Router(config-if)# **tunnel mpls traffic-eng priority**
6. Router(config-if)# **tunnel mpls traffic-eng path-option**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# **interface tunnel1**	Creates a tunnel interface (named in this example **tunnel1**) and enters interface configuration mode.
Step 2	Router(config-if)# **tunnel destination** *A.B.C.D*	Specifies the IP address of the tunnel tail device.
Step 3	Router(config-if)# **tunnel mode mpls traffic-eng**	Sets the tunnel's encapsulation mode to MPLS traffic engineering.
Step 4	Router(config-if)# **tunnel mpls traffic-eng bandwidth** {**sub-pool** | **class-type1**} *bandwidth*	Configures the tunnel's bandwidth, and assigns it either to the sub-pool (when you use that keyword or the IETF-Standard keyword **class-type1**) or to the global pool (when you leave out both keywords).
Step 5	Router(config-if)# **tunnel mpls traffic-eng priority**	Sets the priority to be used when the system determines which existing tunnels are eligible to be preempted.
Step 6	Router(config-if)# **tunnel mpls traffic-eng path-option**	Configures the paths (hops) a tunnel should use. The user can enter an explicit path (can specify the IP addresses of the hops) or can specify a dynamic path (the router figures out the best set of hops).

Verifying the Configuration

To view the complete configuration you have entered, use the EXEC command **show running-config** and check its output display for correctness.

To check just one tunnel's configuration, enter **show interfaces tunnel** followed by the tunnel interface number. And to see that tunnel's RSVP bandwidth and flow, enter **show ip rsvp interface** followed by the name or number of the physical interface.

Here is an example of the information displayed by these latter two commands. (To see an explanation of each field used in the following displays, enter **show interfaces tunnel** or **show ip rsvp interface** into the Command Lookup Tool at http://www.cisco.com/cgi-bin/Support/Cmdlookup/home.pl . If prompted to log in there, use your Cisco.com account username and password.)

```
Router# show interfaces tunnel 4
Tunnel4 is up, line protocol is down
  Hardware is Routing Tunnel
  MTU 1500 bytes, BW 9 Kbit, DLY 500000 usec, rely 255/255, load 1/255
  Encapsulation TUNNEL, loopback not set, keepalive set (10 sec)
  Tunnel source 0.0.0.0, destination 0.0.0.0
  Tunnel protocol/transport GRE/IP, key disabled, sequencing disabled
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Output queue 0/0, 0 drops; input queue 0/75, 0 drops
  Five minute input rate 0 bits/sec, 0 packets/sec
  Five minute output rate 0 bits/sec, 0 packets/sec
0 packets input, 0 bytes, 0 no buffer
Received 0 broadcasts, 0 runts, 0 giants
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
0 packets output, 0 bytes, 0 underruns
0 output errors, 0 collisions, 0 interface resets, 0 restarts
```

```
Router# show ip rsvp interface pos4/0
interface    allocated  i/f max  flow max sub max
PO4/0        300K       466500K  466500K  0M
```

To view all tunnels at once on the router you have configured, enter **show mpls traffic-eng tunnels brief** . The information displayed when tunnels are functioning properly looks like this:

```
Router# show mpls traffic-eng tunnels brief
Signalling Summary:
LSP Tunnels Process:           running
RSVP Process:                  running
Forwarding:                    enabled
Periodic reoptimization:       every 3600 seconds, next in 3029 seconds
TUNNEL NAME  DESTINATION      UP IF     DOWN IF   STATE/PROT
GSR1_t0  192.168.1.13     -          SR3/0     up/up
GSR1_t1  192.168.1.13     -          SR3/0     up/up
GSR1_t2  192.168.1.13     -          PO4/0     up/up
Displayed 3 (of 3) heads, 0 (of 0) midpoints, 0 (of 0) tails
```

When one or more tunnels is not functioning properly, the display could instead look like this. (In the following example, tunnels t0 and t1 are down, as indicated in the far right column).

```
Router# show mpls traffic-eng tunnels brief
Signalling Summary:
    LSP Tunnels Process:           running
    RSVP Process:                  running
    Forwarding:                    enabled
    Periodic reoptimization:       every 3600 seconds, next in 2279 seconds
TUNNEL NAME  DESTINATION      UP IF     DOWN IF   STATE/PROT
GSR1_t0  192.168.1.13     -          SR3/0     up/down
GSR1_t1  192.168.1.13     -          SR3/0     up/down
GSR1_t2  192.168.1.13     -          PO4/0     up/up
Displayed 3 (of 3) heads, 0 (of 0) midpoints, 0 (of 0) tails
```

To find out why a tunnel is down, insert its name into this same command, after adding the keyword **name** and omitting the keyword **brief**. For example:

```
Router# show mpls traffic-eng tunnels name GSR1_t0

Name:GSR1_t0                           (Tunnel0) Destination:192.168.1.13
  Status:
    Admin:up         Oper:down  Path: not valid       Signalling:connected
```

If, as in this example, the Path is displayed as not valid, use the **show mpls traffic-eng topology** command to make sure the router has received the needed updates.

Additionally, you can use any of the following **show** commands to inspect particular aspects of the network, router, or interface concerned:

To see information about...	Use this command	
this level	and this item...	
Network	Advertised bandwidth allocation information	**show mpls traffic-eng link-management advertisements**
	Preemptions along the tunnel path	**debug mpls traffic-eng link-management preemption**
	Available TE link band- width on all head routers	**show mpls traffic-eng topology**

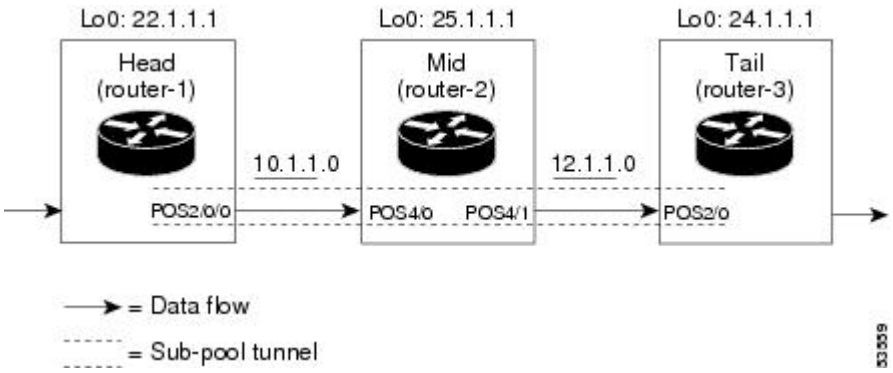
To see information about...	Use this command	
Router	Status of all tunnels cur- rently signalled by this router	**show mpls traffic-eng link-management admission-control**
	Tunnels configured on midpoint routers	**show mpls traffic-eng link-management summary**
Physical interface	Detailed information on current bandwidth pools	**show mpls traffic-eng link-management bandwidth-allocation** [*interface-name*]
	TE RSVP bookkeeping	**show mpls traffic-eng link-management interfaces**
	Entire configuration of one interface	**show run interface**

MPLS Traffic Engineering - DiffServ Aware (DS-TE): Examples

Note The following examples illustrate DS-TE in the traditional, pre-IETF-Standard mode. You may update these examples simply by inserting the new Device Level command **mpls traffic-eng ds-te bc-model** as its proper use is shown in Step 3 on MPLS Traffic Engineering - DiffServ Aware (DS-TE): Examples, on page 13 , and by applying the updated syntax within the two modified commands as each is shown respectively at the Physical Interface Level in Step 2 on MPLS Traffic Engineering - DiffServ Aware (DS-TE): Examples, on page 13 (**ip rsvp bandwidth**), and at the Tunnel Interface Level in Step 4 on MPLS Traffic Engineering - DiffServ Aware (DS-TE): Examples, on page 13 (**tunnel mpls traffic-eng bandwidth**).

First this section presents the DS-TE configurations needed to create the sub-pool tunnel. Then it presents the more comprehensive design for building end-to-end guaranteed bandwidth service, which involves configuring Quality of Service as well.

As shown in the figure below, the tunnel configuration involves at least three devices--tunnel head, midpoint, and tail. On each of those devices one or two network interfaces must be configured, for traffic ingress and egress.



Tunnel Head: Example

At the device level:

```
router-1# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
router-1(config)# ip cef distributed
router-1(config)# mpls traffic-eng tunnels
```

[now one uses either the IS-IS commands on the left or the OSPF commands on the right]:

`router-1(config)# router isis`	`router ospf 100`
`router-1(config-router)# net 49.0000.1000.0000.0010.00`	`redistribute connected`
`router-1(config-router)# metric-style wide`	`network 10.1.1.0 0.0.0.255 area 0`
`router-1(config-router)# is-type level-1`	`network 22.1.1.1 0.0.0.0 area 0`
`router-1(config-router)# mpls traffic-eng level-1`	`mpls traffic-eng area 0`
`router-1(config-router)# passive-interface Loopback0`	

[now one resumes the common command set]:

```
router-1(config-router)# mpls traffic-eng router-id Loopback0
router-1(config-router)# exit
router-1(config)# interface Loopback0
```

At the virtual interface level:

```
router-1(config-if)# ip address 22.1.1.1 255.255.255.255
router-1(config-if)# no ip directed-broadcast
router-1(config-if)# exit
```

At the device level:

```
router-1(config)# interface POS2/0/0
```

At the physical interface level (egress):

```
router-1(config-if)# ip address 10.1.1.1 255.255.255.0
router-1(config-if)# mpls traffic-eng tunnels
router-1(config-if)# ip rsvp bandwidth 130000 130000 sub-pool 80000
[and if using IS-IS instead of OSPF]:
router-1(config-if)# ip router isis
[and in all cases]:
router-1(config-if)# exit
```

At the device level:

```
router-1(config)# interface Tunnel1
```

At the tunnel interface level:

```
router-1(config-if)# bandwidth 110000
router-1(config-if)# ip unnumbered Loopback0
router-1(config-if)# tunnel destination 24.1.1.1
router-1(config-if)# tunnel mode mpls traffic-eng
router-1(config-if)# tunnel mpls traffic-eng priority 0 0
router-1(config-if)# tunnel mpls traffic-eng bandwidth sub-pool 30000
router-1(config-if)# tunnel mpls traffic-eng path-option 1 dynamic
router-1(config-if)# exit
router-1(config)#
```

Midpoint Devices: Example

At the device level:

```
router-2# configure terminal
router-2(config)# ip cef distributed
router-2(config)# mpls traffic-eng tunnels
```

[now one uses either the IS-IS commands on the left or the OSPF commands on the right]:

`router-2(config)# router isis`	`router ospf 100`
`router-2(config-router)# net 49.0000.1000.0000.0012.00`	`redistribute connected`
`router-2(config-router)# metric-style wide`	`network 11.1.1.0 0.0.0.255 area 0`
`router-2(config-router)# is-type level-1`	`network 12.1.1.0 0.0.0.255 area 0`
`router-2(config-router)# mpls traffic-eng level-1`	`network 25.1.1.1 0.0.0.0 area 0`
`router-2(config-router)# passive-interface Loopback0`	`mpls traffic-eng area 0`

[now one resumes the common command set]:

```
router-2(config-router)# mpls traffic-eng router-id Loopback0
router-2(config-router)# exit
router-2(config)# interface Loopback0
```

At the virtual interface level:

```
router-2(config-if)# ip address 25.1.1.1 255.255.255.255
router-2(config-if)# no ip directed-broadcast
router-2(config-if)# exit
```

```
At the device level:
router-1(config)# interface POS4/0
router-1(config-if)# ip address 11.1.1.2 255.255.255.0
router-1(config-if)# mpls traffic-eng tunnels
router-1(config-if)# ip rsvp bandwidth 130000 130000 sub-pool 80000
```

[If using IS-IS instead of OSPF]:

```
router-1(config-if)# ip router isis
[and in all cases]:
router-1(config-if)# exit
At the device level:
router-1(config)# interface POS4/1
router-1(config-if)# ip address 12.1.1.2 255.255.255.0
router-1(config-if)# mpls traffic-eng tunnels
router-1(config-if)# ip rsvp bandwidth 130000 130000 sub-pool 80000
```

[If using IS-IS instead of OSPF]:

```
router-1(config-if)# ip router isis
[and in all cases]:
router-1(config-if)# exit
```

Note that there is no configuring of tunnel interfaces at the mid-point devices, only network interfaces and the device globally.

Tail-End Device: Example

At the device level:

```
router-3# configure terminal
router-3(config)# ip cef distributed
router-3(config)# mpls traffic-eng tunnels
```

[now one uses either the IS-IS commands on the left or the OSPF commands on the right]:

`router-3(config)# router isis`	`router ospf 100`
`router-3(config-router)# net 49.0000.1000.0000.0013.00`	`redistribute connected`
`router-3(config-router)# metric-style wide`	`network 12.1.1.0 0.0.0.255 area 0`
`router-3(config-router)# is-type level-1`	`network 24.1.1.1 0.0.0.0 area 0`
`router-3(config-router)# mpls traffic-eng level-1`	`mpls traffic-eng area 0`
`router-3(config-router)# passive-interface Loopback0`	

[now one resumes the common command set]:

```
router-3(config-router)# mpls traffic-eng router-id Loopback0
router-3(config-router)# exit
router-3(config)# interface Loopback0
```

At the virtual interface level:

```
router-3(config-if)# ip address 24.1.1.1 255.255.255.255
router-3(config-if)# no ip directed-broadcast
[and if using IS-IS instead of OSPF]:
router-3(config-if)# ip router isis
[and in all cases]:
router-3(config-if)# exit
```

At the device level:

```
router-1(config)# interface POS4/0
router-1(config-if)# ip address 12.1.1.3 255.255.255.0
router-1(config-if)# mpls traffic-eng tunnels
router-1(config-if)# ip rsvp bandwidth 130000 130000 sub-pool 80000
```

[If using IS-IS instead of OSPF]:

```
router-1(config-if)# ip router isis
[and in all cases]:
router-1(config-if)# exit
```

Guaranteed Bandwidth Service: Examples

Given the many topologies in which Guaranteed Bandwidth Services can be applied, there is space here only to present two examples. They illustrate opposite ends of the spectrum of possibilities.

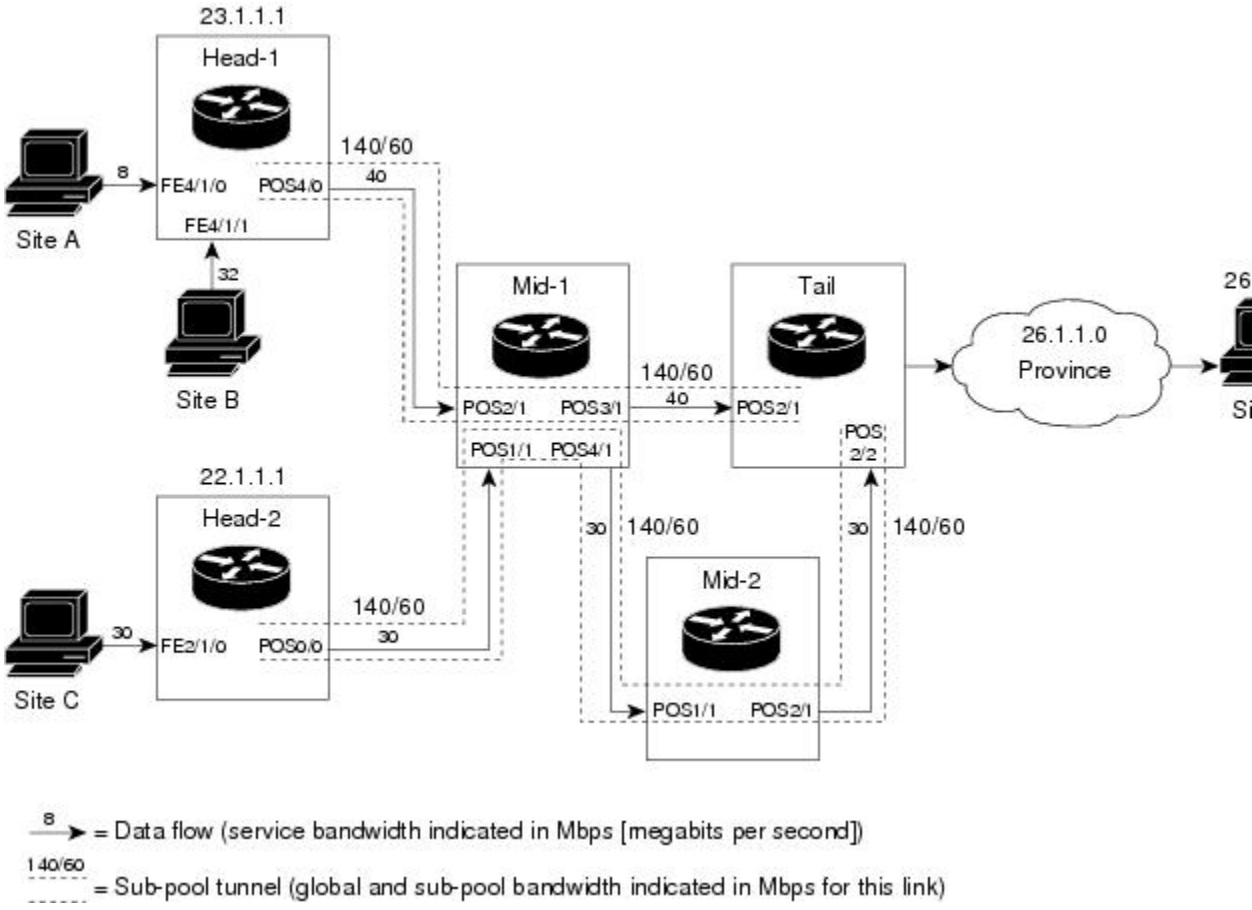
In the first example, the guaranteed bandwidth tunnel can be easily specified by its destination. So the forwarding criteria refer to a single destination prefix.

In the second example, there can be many final destinations for the guaranteed bandwidth traffic, including a dynamically changing number of destination prefixes. So the forwarding criteria are specified by Border Gateway Protocol (BGP) policies.

Single Destination Prefix: Example

The figure below illustrates atopology for guaranteed bandwidth services whose destination is specified by a single prefix, either Site D (like a voice gateway, here bearing prefix 26.1.1.1) or a subnet (like the location of a web farm, here called “Province” and bearing prefix 26.1.1.0). Three services are offered:

- From Site A (defined as all traffic arriving at interface FE4/1/0): to host 26.1.1.1, 8 Mbps of guaranteed bandwidth with low loss, low delay and low jitter
- From Site B (defined as all traffic arriving at interface FE4/1/1): towards subnet 26.1.1.0, 32 Mbps of guaranteed bandwidth with low loss
- From Site C (defined as all traffic arriving at interface FE2/1/0): 30 Mbps of guaranteed bandwidth with low loss



These three services run through two sub-pool tunnels:

- From the Head-1 router, 23.1.1.1, to the router-4 tail
- From the Head-2 router, 22.1.1.1, to the router-4 tail

Both tunnels use the same tail router, though they have different heads. (In the figure above one midpoint router is shared by both tunnels. In the real world there could of course be many more midpoints.)

All POS interfaces in this example are OC3, whose capacity is 155 Mbps.

Configuring Tunnel Head-1 Example

First we recapitulate commands that establish two bandwidth pools and a sub-pool tunnel (as presented earlier in this Configuration Examples section). Then we present the QoS commands that guarantee end-to-end service on the subpool tunnel. (With the 7500 router, Modular QoS CLI is used.)

At the device level:

```
router-1(config)# ip cef distributed
router-1(config)# mpls traffic-eng tunnels
```

[now one uses either the IS-IS commands on the left or the OSPF commands on the right]:

router-1(config)# **router isis**	**router ospf 100**
router-1(config-router)# **net 49.0000.1000.0000.0010.00**	**redistribute connected**
router-1(config-router)# **metric-style wide**	**network 10.1.1.0 0.0.0.255 area 0**
router-1(config-router)# **is-type level-1**	**network 23.1.1.1 0.0.0.0 area 0**
router-1(config-router)# **mpls traffic-eng level-1**	**mpls traffic-eng area 0**
router-1(config-router)# **passive-interface Loopback0**	

[now one resumes the common command set]:

```
router-1(config-router)# mpls traffic-eng router-id Loopback0
router-1(config-router)# exit
```

Create a virtual interface:

```
router-1(config)# interface Loopback0
router-1(config-if)# ip address 23.1.1.1 255.255.255.255
router-1(config-if)# no ip directed-broadcast
router-1(config-if)# exit
```

At the outgoing physical interface:

```
router-1(config)# interface pos4/0
router-1(config-if)# ip address 10.1.1.1 255.0.0.0
router-1(config-if)# mpls traffic-eng tunnels
router-1(config-if)# ip rsvp bandwidth 140000 140000 sub-pool 60000
[and if using IS-IS instead of OSPF]:
router-1(config-if)# ip router isis
[and in all cases}:
router-1(config-if)# exit
```

At the tunnel interface:

```
router-1(config)# interface Tunnel1
router-1(config-if)# bandwidth 110000
router-1(config-if)# ip unnumbered Loopback0
router-1(config-if)# tunnel destination 27.1.1.1
router-1(config-if)# tunnel mode mpls traffic-eng
router-1(config-if)# tunnel mpls traffic-eng priority 0 0
router-1(config-if)# tunnel mpls traffic-eng bandwidth sub-pool 40000
router-1(config-if)# tunnel mpls traffic-eng path-option 1 dynamic
```

To ensure that packets destined to host 26.1.1.1 and subnet 26.1.1.0 are sent into the sub-pool tunnel, we create a static route. At the device level:

```
router-1(config)# ip route 26.1.1.0 255.255.255.0 Tunnel1
router-1(config)# exit
```

And in order to make sure that the Interior Gateway Protocol (IGP) will not send any other traffic down this tunnel, we disable autoroute announce:

```
router-1(config)# no tunnel mpls traffic-eng autoroute announce
```

At the inbound physical interface (FE4/1/0):

1 In global configuration mode, create a class of traffic matching ACL 100, called "sla-1-class":

```
class-map match-all sla-1-class
match access-group 100
```

1 Create an ACL 100 to refer to all packets destined to 26.1.1.1:

```
access-list 100 permit ip any host 26.1.1.1
```

1 Create a policy named “sla-1-input-policy”, and according to that policy:

  1 Packets in the class called “sla-1-class” are rate-limited to:

- a rate of 8 million bits per second
- a normal burst of 1 million bytes
- a maximum burst of 2 million bytes

1 Packets which conform to this rate are marked with MPLS experimental bit 5 and are forwarded.

2 Packets which exceed this rate are dropped.

3 All other packets are marked with experimental bit 0 and are forwarded.

```
policy-map sla-1-input-policy
class sla-1-class
police 8000000 1000000 2000000 conform-action set-mpls-exp-transmit 5 \ exceed-action drop
class class-default
set-mpls-exp-transmit 0
```

1 The policy is applied to packets entering interface FE4/1/0.

```
interface FastEthernet4/1/0
service-policy input sla-1-input-policy
```

At the inbound physical interface (FE4/1/1):

1 In global configuration mode, create a class of traffic matching ACL 120, called "sla-2-class":

```
class-map match-all sla-2-class
match access-group 120
```

1 Create an ACL, 120, to refer to all packets destined to subnet 26.1.1.0:

```
access-list 120 permit ip any 26.1.1.0 0.0.0.255
```

1 Create a policy named “sla-2-input-policy”, and according to that policy:

  1 Packets in the class called “sla-2-class” are rate-limited to:

- a rate of 32 million bits per second

- a normal burst of 1 million bytes

- a maximum burst of 2 million bytes

1. Packets which conform to this rate are marked with MPLS experimental bit 5 and are forwarded.
2. Packets which exceed this rate are dropped.
3. All other packets are marked with experimental bit 0 and are forwarded.

```
policy-map sla-2-input-policy
class sla-2-class
police 32000000 1000000 2000000 conform-action set-mpls-exp-transmit 5 \ exceed-action drop
class class-default
set-mpls-exp-transmit 0
```

1. The policy is applied to packets entering interface FE4/1/1.

```
interface FastEthernet4/1/1
service-policy input sla-2-input-policy
```

The outbound interface (POS4/0) is configured as follows:

1. In global configuration mode, create a class of traffic matching experimental bit 5, called "exp-5-traffic".

```
class-map match-all exp-5-traffic
match mpls experimental 5
```

1. Create a policy named “output-interface-policy”. According to that policy, packets in the class “exp-5-traffic” are put in the priority queue (which is rate-limited to 62 kbits/sec).

```
policy-map output-interface-policy
class exp-5-traffic
priority 32
```

1. The policy is applied to packets exiting interface POS4/0.

```
   interface POS4/0
service-policy output output-interface-policy
```

The result of the above configuration lines is that packets entering the Head-1 router via interface FE4/1/0 destined to host 26.1.1.1, or entering the router via interface FE4/1/1 destined to subnet 26.1.1.0, will have their MPLS experimental bit set to 5. We assume that no other packets entering the router (on any interface) are using this value. (If this cannot be assumed, an additional configuration must be added to mark all such packets to another experimental value.) Packets marked with experimental bit 5, when exiting the router via interface POS4/0, will be placed into the priority queue.

Note Packets entering the router via FE4/1/0 or FE4/1/1 and exiting POS4/0 enter as IP packets and exit as MPLS packets.

Configuring Tunnel Head-2 Example

First we recapitulate commands that establish two bandwidth pools and a sub-pool tunnel (as presented earlier in this Configuration Examples section). Then we present the QoS commands that guarantee end-to-end service on the sub-pool tunnel.

At the device level:

```
router-2(config)# ip cef distributed
router-2(config)# mpls traffic-eng tunnels
```

[now one uses either the IS-IS commands on the left or the OSPF commands on the right]:

`router-2(config)# router isis`	`router ospf 100`
`router-2(config-router)# net 49.0000.1000.0000.0011.00`	`redistribute connected`
`router-2(config-router)# metric-style wide`	`network 11.1.1.0 0.0.0.255 area 0`
`router-2(config-router)# is-type level-1`	`network 22.1.1.1 0.0.0.0 area 0`
`router-2(config-router)# mpls traffic-eng level-1`	`mpls traffic-eng area 0`
`router-2(config-router)# passive-interface Loopback0`	

[now one resumes the common command set]:

```
router-2(config-router)# mpls traffic-eng router-id Loopback0
router-2(config-router)# exit
```

Create a virtual interface:

```
router-2(config)# interface Loopback0
router-2(config-if)# ip address 22.1.1.1 255.255.255.255
router-2(config-if)# no ip directed broadcast
router-2(config-if)# exit
```

At the outgoing physical interface:

```
router-2(config)# interface pos0/0
router-2(config-if)# ip address 11.1.1.1 255.0.0.0
router-2(config-if)# mpls traffic-eng tunnels
router-2(config-if)# ip rsvp bandwidth 140000 140000 sub-pool 60000
[and if using IS-IS instead of OSPF]:
router-2(config-if)# ip router isis
[and in all cases]:
router-2(config-if)# exit
```

At the tunnel interface:

```
router-2(config)# interface Tunnel2
router-2(config-if)# ip unnumbered Loopback0
```

```
router-2(config-if)# tunnel destination 27.1.1.1
router-2(config-if)# tunnel mode mpls traffic-eng
router-2(config-if)# tunnel mpls traffic-eng priority 0 0
router-2(config-if)# tunnel mpls traffic-eng bandwidth sub-pool 30000
router-2(config-if)# tunnel mpls traffic-eng path-option 1 dynamic
router-2(config-if)# exit
```

And to ensure that packets destined to subnet 26.1.1.0 are sent into the sub-pool tunnel, we create a static route, at the device level:

```
router-2(config)# ip route 26.1.1.0 255.255.255.0 Tunnel2
router-2(config)# exit
```

Finally, in order to make sure that the Interior Gateway Protocol (IGP) will not send any other traffic down this tunnel, we disable autoroute announce:

```
router-2(config)# no tunnel mpls traffic-eng autoroute announce
 At the inbound physical interface (FE2/1/0):
```

1 In global configuration mode, create a class of traffic matching ACL 130, called "sla-3-class":

```
class-map match-all sla-3-class
match access-group 130
```

1 Create an ACL, 130, to refer to all packets destined to subnet 26.1.1.0:

```
access-list 130 permit ip any 26.1.1.0 0.0.0.255
```

1 Create a policy named “sla-3-input-policy”, and according to that policy:

1 Packets in the class called “sla-3-class” are rate-limited to:

- a rate of 30 million bits per second

- a normal burst of 1 million bytes

- a maximum burst of 2 million bytes

1 Packets which conform to this rate are marked with MPLS experimental bit 5 and are forwarded.

2 Packets which exceed this rate are dropped.

3 All other packets are marked with experimental bit 0 and are forwarded.

```
policy-map sla-3-input-policy
class sla-3-class
police 30000000 1000000 2000000 conform-action set-mpls-exp-transmit 5 \ exceed-action drop
class class-default
set-mpls-exp-transmit 0
```

1 The policy is applied to packets entering interface FE2/1/0.

```
interface FastEthernet2/1/0
service-policy input sla-3-input-policy
```

The outbound interface POS0/0 is configured as follows:

1 In global configuration mode, create a class of traffic matching experimental bit 5, called "exp-5-traffic".

```
class-map match-all exp-5-traffic
match mpls experimental 5
```

1 Create a policy named “output-interface-policy”. According to that policy, packets in the class “exp-5-traffic” are put in the priority queue (which is rate-limited to 32 kbits/sec).

```
policy-map output-interface-policy
class exp-5-traffic
priority 32
```

1 The policy is applied to packets exiting interface POS0/0:

```
interface POS0/0
service-policy output output-interface-policy
```

As a result of all the above configuration lines, packets entering the Head-2 router via interface FE2/1/0 and destined for subnet 26.1.1.0 have their IP precedence field set to 5. It is assumed that no other packets entering this router (on any interface) are using this precedence. (If this cannot be assumed, an additional configuration must be added to mark all such packets with another precedence value.) When exiting this router via interface POS0/0, packets marked with precedence 5 are placed in the priority queue.

Note Packets entering the router via FE2/1/0 and exiting through POS0/0 enter as IP packets and exit as MPLS packets.

Tunnel Midpoint Configuration Mid-1 Example

All four interfaces on the midpoint router are configured identically to the outbound interface of the head router (except, of course, for the IDs of the individual interfaces):

Configuring the Pools and Tunnels

At the device level:

```
router-3(config)# ip cef distributed
router-3(config)# mpls traffic-eng tunnels
```

[now one uses either the IS-IS commands on the left or the OSPF commands on the right]:

`router-3(config)# router isis`	`router ospf 100`
`router-3(config-router)# net 49.0000.2400.0000.0011.00`	`redistribute connected`
`router-3(config-router)# metric-style wide`	`network 10.1.1.0 0.0.0.255 area 0`
`router-3(config-router)# is-type level-1`	`network 11.1.1.0 0.0.0.255 area 0`
`router-3(config-router)# mpls traffic-eng level-1`	`network 24.1.1.1 0.0.0.0 area 0`
`router-3(config-router)# passive-interface Loopback0`	`network 12.1.1.0 0.0.0.255 area 0`
`router-3(config-router)#`	`network 13.1.1.0 0.0.0.255 area 0`
`router-3(config-router)#`	`mpls traffic-eng area 0`

[now one resumes the common command set]:

```
router-3(config-router)# mpls traffic-eng router-id Loopback0
router-3(config-router)# exit
```

Create a virtual interface:

```
router-3(config)# interface Loopback0
router-3(config-if)# ip address 24.1.1.1 255.255.255.255
router-3(config-if)# exit
```

At the physical interface level (ingress):

```
router-3(config)# interface pos2/1
router-3(config-if)# ip address 10.1.1.2 255.0.0.0
router-3(config-if)# mpls traffic-eng tunnels
router-3(config-if)# ip rsvp bandwidth 140000 140000 sub-pool 60000
[and if using IS-IS instead of OSPF]:
router-3(config-if)# ip router isis
[and in all cases]:
router-3(config-if)# exit
router-3(config)# interface pos1/1
router-3(config-if)# ip address 11.1.1.2 255.0.0.0
router-3(config-if)# mpls traffic-eng tunnels
router-3(config-if)# ip rsvp bandwidth 140000 140000 sub-pool 60000
[and if using IS-IS instead of OSPF]:
router-3(config-if)# ip router isis
[and in all cases]:
router-3(config-if)# exit
```

At the physical interface level (egress):

```
router-3(config)# interface pos3/1
```

```
router-3(config-if)# ip address 12.1.1.1 255.0.0.0
router-3(config-if)# mpls traffic-eng tunnels
router-3(config-if)# ip rsvp bandwidth 140000 140000 sub-pool 60000
[and if using IS-IS instead of OSPF]:
router-3(config-if)# ip router isis
[and in all cases]:
router-3(config-if)# exit
router-3(config)# interface pos4/1
router-3(config-if)# ip address 13.1.1.1 255.0.0.0
router-3(config-if)# mpls traffic-eng tunnels
router-3(config-if)# ip rsvp bandwidth 140000 140000 sub-pool 60000
[and if using IS-IS instead of OSPF]:
router-3(config-if)# ip router isis
[and in all cases]:
router-3(config-if)# exit
```

Tunnel Midpoint Configuration Mid-2 Example

Both interfaces on the midpoint router are configured identically to the outbound interface of the head router (except, of course, for the IDs of the individual interfaces):

At the device level:

```
router-5(config)# ip cef distributed
router-5(config)# mpls traffic-eng tunnels
```

[now one uses either the IS-IS commands on the left or the OSPF commands on the right]:

`router-5(config)# router isis`	`router ospf 100`
`router-5(config-router)# net 49.2500.1000.0000.0012.00`	`redistribute connected`
`router-5(config-router)# metric-style wide`	`network 13.1.1.0 0.0.0.255 area 0`
`router-5(config-router)# is-type level-1`	`network 14.1.1.0 0.0.0.255 area 0`
`router-5(config-router)# mpls traffic-eng level-1`	`network 25.1.1.1 0.0.0.0 area 0`
`router-5(config-router)# passive-interface Loopback0`	`mpls traffic-eng area 0`

[now one resumes the common command set]:

```
router-5(config-router)# mpls traffic-eng router-id Loopback0
router-5(config-router)# exit
```

Create a virtual interface:

```
router-5(config)# interface Loopback0
router-5(config-if)# ip address 25.1.1.1 255.255.255.255
router-5(config-if)# exit
```

At the physical interface level (ingress):

```
router-5(config)# interface pos1/1
router-5(config-if)# ip address 13.1.1.2 255.0.0.0
router-5(config-if)# mpls traffic-eng tunnels
router-5(config-if)# ip rsvp bandwidth 140000 140000 sub-pool 60000
[and if using IS-IS instead of OSPF]:
router-5(config-if)# ip router isis
[and in all cases]:
router-5(config-if)# exit
```

At the physical interface level (egress):

```
router-5(config)# interface pos2/1
router-5(config-if)# ip address 14.1.1.1 255.0.0.0
router-5(config-if)# mpls traffic-eng tunnels
router-5(config-if)# ip rsvp bandwidth 140000 140000 sub-pool 60000
[and if using IS-IS instead of OSPF]:
router-5(config-if)# ip router isis
[and in all cases]:
router-5(config-if)# exit
```

Tunnel Tail Configuration Example

The inbound interfaces on the tail router are configured identically to the inbound interfaces of the midpoint routers (except, of course, for the ID of each particular interface):

At the device level:

```
router-4(config)# ip cef distributed
router-4(config)# mpls traffic-eng tunnels
```

[now one uses either the IS-IS commands on the left or the OSPF commands on the right]:

`router-4(config)# router isis`	`router ospf 100`
`router-4(config-router)# net 49.0000.2700.0000.0000.00`	`redistribute connected`
`router-4(config-router)# metric-style wide`	`network 12.1.1.0 0.0.0.255 area 0`
`router-4(config-router)# is-type level-1`	`network 14.1.1.0 0.0.0.255 area 0`
`router-4(config-router)# mpls traffic-eng level-1`	`network 27.1.1.1 0.0.0.0 area 0`
`router-4(config-router)# passive-interface Loopback0`	`mpls traffic-eng area 0`

[now one resumes the common command set]:

```
router-4(config-router)# mpls traffic-eng router-id Loopback0
router-4(config-router)# exit
```

Create a virtual interface:

```
router-4(config)# interface Loopback0
router-4(config-if)# ip address 27.1.1.1 255.255.255.255
router-4(config-if)# exit
```

At the physical interface (ingress):

```
router-4(config)# interface pos2/1
router-4(config-if)# ip address 12.1.1.2 255.0.0.0
router-4(config-if)# mpls traffic-eng tunnels
router-4(config-if)# ip rsvp bandwidth 140000 140000 sub-pool 60000
[and if using IS-IS instead of OSPF]:
router-4(config-if)# ip router isis
[and in all cases]:
router-4(config-if)# exit
router-4(config)# interface pos2/2
router-4(config-if)# ip address 14.1.1.2 255.0.0.0
router-4(config-if)# mpls traffic-eng tunnels
router-4(config-if)# ip rsvp bandwidth 140000 140000 sub-pool 60000
[and if using IS-IS instead of OSPF]:
router-4(config-if)# ip router isis
[and in all cases]:
router-4(config-if)# exit
```

Because the tunnel ends on the tail (does not include any outbound interfaces of the tail router), no outbound QoS configuration is used.

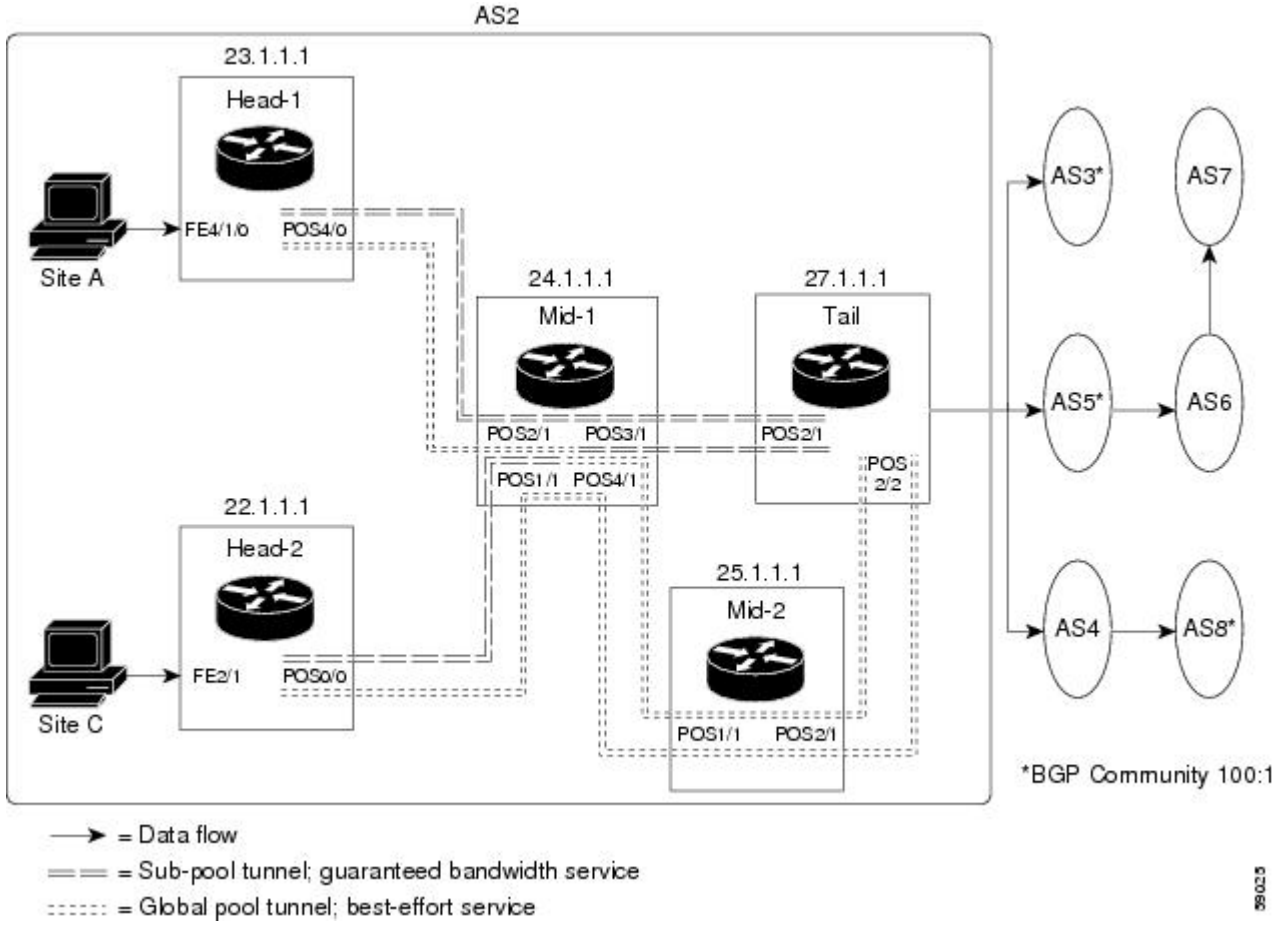
Many Destination Prefixes: Example

The figure below illustrates a topology for guaranteed bandwidth services whose destinations are a set of prefixes. Those prefixes usually share some common properties such as belonging to the same Autonomous System (AS) or transiting through the same AS. Although the individual prefixes may change dynamically because of route flaps in the downstream autonomous systems, the properties the prefixes share will not change. Policies addressing the destination prefix set are enforced through Border Gateway Protocol (BGP), which is described in the following documents:

- "Configuring QoS Policy Propagation via Border Gateway Protocol" in the *Cisco IOS Quality of Service Solutions Configuration Guide*
- "Configuring BGP" in the *Cisco IOS IP and IP Routing Configuration Guide*
- "BGP Commands" in the *Cisco IOS IP and IP Routing Command Reference*
- "BGP-Policy Command" in the *Cisco IOS Quality of Service Solutions Command Reference*

In this example, three guaranteed bandwidth services are offered, each coming through a 7500 or a 12000 edge device:

- Traffic coming from Site A (defined as all traffic arriving at interface FE4/1/0) and from Site C (defined as all traffic arriving at interface FE2/1) destined to AS5
- Traffic coming from Sites A and C that transits AS5 but is not destined to AS5. (In the figure, the transiting traffic will go to AS6 and AS7)
- Traffic coming from Sites A and C destined to prefixes advertised with a particular BGP community attribute (100:1). In this example, Autonomous Systems #3, #5, and #8 are the BGP community assigned the attribute 100:1.



The applicability of guaranteed bandwidth service is not limited to the three types of multiple destination scenarios described above. There is not room in this document to present all possible scenarios. These three were chosen as representative of the wide range of possible deployments.

The guaranteed bandwidth services run through two sub-pool tunnels:

- From the Head-1 router, 23.1.1.1, to the tail
- From the Head-2 router, 22.1.1.1, to that same tail

In addition, a global pool tunnel has been configured from each head end, to carry best-effort traffic to the same destinations. All four tunnels use the same tail router, even though they have different heads and differ in their passage through the midpoints. (Of course in the real world there would be many more midpoints than just the two shown here.)

All POS interfaces in this example are OC3, whose capacity is 155 Mbps.

Configuring a multi-destination guaranteed bandwidth service involves:

1. Building a sub-pool MPLS-TE tunnel
2. Configuring DiffServ QoS
3. Configuring QoS Policy Propagation via BGP (QPPB)
4. Mapping traffic onto the tunnels

All of these tasks are included in the following example.

Configuration of Tunnel Head-1 Example

First we recapitulate commands that establish a sub-pool tunnel (commands presented earlier in MPLS Traffic Engineering - DiffServ Aware (DS-TE): Examples, on page 13) and now we also configure a global pool tunnel. Additionally, we present QoS and BGP commands that guarantee end-to-end service on the sub-pool tunnel. (With the 7500(VIP) router, Modular QoS CLI is used).

At the device level:

```
router-1(config)# ip cef distributed
router-1(config)# mpls traffic-eng tunnels
```

[now one uses either the IS-IS commands on the left or the OSPF commands on the right]:

`router-1(config)# router isis`	`router ospf 100`
`router-1(config-router)# net 49.0000.1000.0000.0010.00`	`redistribute connected`
`router-1(config-router)# metric-style wide`	`network 10.1.1.0 0.0.0.255 area 0`
`router-1(config-router)# is-type level-1`	`network 23.1.1.1 0.0.0.0 area 0`
`router-1(config-router)# mpls traffic-eng level-1`	`mpls traffic-eng area 0`

[now one resumes the common command set]:

```
router-1(config-router)# mpls traffic-eng router-id Loopback0
router-1(config-router)# exit
```

Create a virtual interface:

```
router-1(config)# interface Loopback0
router-1(config-if)# ip address 23.1.1.1 255.255.255.255
router-1(config-if)# exit
```

At the outgoing physical interface:

```
router-1(config)# interface pos4/0
router-1(config-if)# ip address 10.1.1.1 255.0.0.0
router-1(config-if)# mpls traffic-eng tunnels
router-1(config-if)# ip rsvp bandwidth 140000 140000 sub-pool 60000
[and if using IS-IS instead of OSPF]:
router-1(config-if)# ip router isis
[and in all cases]:
router-1(config-if)# exit
```

At one tunnel interface, create a sub-pool tunnel:

```
router-1(config)# interface Tunnel1
router-1(config-if)# ip unnumbered Loopback0
router-1(config-if)# tunnel destination 27.1.1.1
```

```
router-1(config-if)# tunnel mode mpls traffic-eng
router-1(config-if)# tunnel mpls traffic-eng priority 0 0
router-1(config-if)# tunnel mpls traffic-eng bandwidth sub-pool 40000
router-1(config-if)# tunnel mpls traffic-eng path-option 1 explicit name gbs-path1
router-1(config-if)# exit
```

and at a second tunnel interface, create a global pool tunnel:

```
router-1(config)# interface Tunnel2
router-1(config-if)# ip unnumbered Loopback0
router-1(config-if)# tunnel destination 27.1.1.1
router-1(config-if)# tunnel mode mpls traffic-eng
router-1(config-if)# tunnel mpls traffic-eng priority 0 0
router-1(config-if)# tunnel mpls traffic-eng bandwidth 80000
router-1(config-if)# tunnel mpls traffic-eng path-option 1 explicit name \ best-effort-path1
router-1(config-if)# exit
```

In this example explicit paths are used instead of dynamic, to ensure that best-effort traffic and guaranteed bandwidth traffic will travel along different paths.

At the device level:

```
router-1(config)# ip explicit-path name gbs-path1
router-1(config-ip-expl-path)# next-address 24.1.1.1
router-1(config-ip-expl-path)# next-address 27.1.1.1
router-1(config-ip-expl-path)# exit
router-1(config)# ip explicit-path name best-effort-path1
router-1(config-ip-expl-path)# next-address 24.1.1.1
router-1(config-ip-expl-path)# next-address 25.1.1.1
router-1(config-ip-expl-path)# next-address 27.1.1.1
router-1(config-ip-expl-path)# exit
```

Note that autoroute is not used, as that could cause the Interior Gateway Protocol (IGP) to send other traffic down these tunnels.

At the inbound physical interface (in the figure above this is FE4/1/0), packets received are rate-limited to:

1. a rate of 30 Mbps
2. a normal burst of 1 MB
3. a maximum burst of 2 MB

Packets that are mapped to qos-group 6 and that conform to the rate-limit are marked with experimental value 5 and the BGP destination community string, and are forwarded; packets that do not conform (exceed action) are dropped:

```
router-1(config)# interface FastEthernet4/1/0
router-1(config-if)# rate-limit input qos-group 6 30000000 1000000 2000000 \
 conform-action set-mpls-exp-transmit 5 exceed-action drop
router-1(config-if)# bgp-policy destination ip-qos-map
router-1(config-if)# exit
```

At the device level create a class of traffic called “exp5-class” that has MPLS experimental bit set to 5:

```
router-1(config)# class-map match-all exp5-class
router-1(config-cmap)# match mpls experimental 5
router-1(config-cmap)# exit
```

Create a policy that creates a priority queue for “exp5-class”:

```
router-1(config)# policy-map core-out-policy
router-1(config-pmap)# class exp5-class
router-1(config-pmap-c)# priority 100000
router-1(config-pmap-c)# exit
router-1(config-pmap)# class class-default
router-1(config-pmap-c)# bandwidth 55000
```

```
router-1(config-pmap-c)# exit
router-1(config-pmap)# exit
```
The policy is applied to packets exiting the outbound interface POS4/0.

```
router-1(config)# interface POS4/0
router-1(config-if)# service-policy output core-out-policy
```
Create a table map under BGP to map (tie) the prefixes to a qos-group. At the device level:

```
router-1(config)# ip bgp-community new-format
router-1(config)# router bgp 2
router-1(config-router)# no synchronization
router-1(config-router)# table-map set-qos-group
router-1(config-router)# bgp log-neighbor-changes
router-1(config-router)# neighbor 27.1.1.1 remote-as 2
router-1(config-router)# neighbor 27.1.1.1 update-source Loopback0
router-1(config-router)# no auto-summary
router-1(config-router)# exit
```
Create a distinct route map for this service. This includes setting the next-hop of packets matching 29.1.1.1 so they will be mapped onto Tunnel #1 (the guaranteed bandwidth service tunnel). At the device level:

```
router-1(config)# route-map set-qos-group permit 10
router-1(config-route-map)# match as-path 100
router-1(config-route-map)# set ip qos-group 6
router-1(config-route-map)# set ip next-hop 29.1.1.1
router-1(config-route-map)# exit
router-1(config)# ip as-path access-list 100 permit ^5$
```
Create a distinct route map for this service. (Its traffic will go to AS6 and AS7).

At the device level:

```
router-1(config)# route-map set-qos-group permit 10
router-1(config-route-map)# match as-path 101
router-1(config-route-map)# set ip qos-group 6
router-1(config-route-map)# set ip next-hop 29.1.1.1
router-1(config-route-map)# exit
router-1(config)# ip as-path access-list 101 permit _5_
```
Create a distinct route map for all traffic destined to prefixes that have community value 100:1. This traffic will go to AS3, AS5, and AS8.

At the device level:

```
router-1(config)# route-map set-qos-group permit 10
router-1(config-route-map)# match community 20
router-1(config-route-map)# set ip qos-group 6
router-1(config-route-map)# set ip next-hop 29.1.1.1
router-1(config-route-map)# exit
router-1(config)# ip community-list 20 permit 100:1
```
Map all guaranteed bandwidth traffic onto Tunnel #1:

```
router-1(config)# ip route 29.1.1.1 255.255.255.255 Tunnel1
```
Map all best-effort traffic onto Tunnel #2:

```
router-1(config)# ip route 30.1.1.1 255.255.255.255 Tunnel2
```

Configuration of Tunnel Head-2 Example

As with the Head-1 device and interfaces, the following Head-2 configuration first presents commands that establish a sub-pool tunnel (commands presented earlier in MPLS Traffic Engineering - DiffServ Aware (DS-TE): Examples, on page 13) and then also configures a global pool tunnel. After that it presents QoS

and BGP commands that guarantee end-to-end service on the sub-pool tunnel. (Because this is a 7500 (VIP) router, Modular QoS CLI is used).

At the device level:

```
router-2(config)# ip cef distributed
router-2(config)# mpls traffic-eng tunnels
```

[now one uses either the IS-IS commands on the left or the OSPF commands on the right]:

`router-2(config)# router isis`	`router ospf 100`
`router-2(config-router)# net 49.0000.1000.0000.0011.00`	`redistribute connected`
`router-2(config-router)# metric-style wide`	`network 11.1.1.0 0.0.0.255 area 0`
`router-2(config-router)# is-type level-1`	`network 22.1.1.1 0.0.0.0 area 0`
`router-2(config-router)# mpls traffic-eng level-1`	`mpls traffic-eng area 0`

[now one resumes the common command set]:

```
router-2(config-router)# mpls traffic-eng router-id Loopback0
router-2(config-router)# exit
```

Create a virtual interface:

```
router-2(config)# interface Loopback0
router-2(config-if)# ip address 22.1.1.1 255.255.255.255
router-2(config-if)# exit
```

At the outgoing physical interface:

```
router-2(config)# interface pos0/0
router-2(config-if)# ip address 11.1.1.1 255.0.0.0
router-2(config-if)# mpls traffic-eng tunnels
router-2(config-if)# ip rsvp bandwidth 140000 140000 sub-pool 60000
[and if using IS-IS instead of OSPF]:
router-2(config-if)# ip router isis
[and in all cases]:
router-2(config-if)# exit
```

At one tunnel interface, create a sub-pool tunnel:

```
router-2(config)# interface Tunnel3
router-2(config-if)# ip unnumbered Loopback0
router-2(config-if)# tunnel destination 27.1.1.1
router-2(config-if)# tunnel mode mpls traffic-eng
router-2(config-if)# tunnel mpls traffic-eng priority 0 0
router-2(config-if)# tunnel mpls traffic-eng bandwidth sub-pool 30000
router-2(config-if)# tunnel mpls traffic-eng path-option 1 explicit name gbs-path2
router-2(config-if)# exit
```

and at a second tunnel interface, create a global pool tunnel:

```
router-2(config)# interface Tunnel4
router-2(config-if)# ip unnumbered Loopback0
router-2(config-if)# tunnel destination 27.1.1.1
router-2(config-if)# tunnel mode mpls traffic-eng
router-2(config-if)# tunnel mpls traffic-eng priority 0 0
router-2(config-if)# tunnel mpls traffic-eng bandwidth 70000
router-2(config-if)# tunnel mpls traffic-eng path-option 1 explicit name \ best-effort-path2
router-2(config-if)# exit
```

In this example explicit paths are used instead of dynamic, to ensure that best-effort traffic and guaranteed bandwidth traffic will travel along different paths.

At the device level:

```
router-2(config)# ip explicit-path name gbs-path2
router-2(config-ip-expl-path)# next-address 24.1.1.1
router-2(config-ip-expl-path)# next-address 27.1.1.1
router-2(config-ip-expl-path)# exit
router-2(config)# ip explicit-path name best-effort-path2
router-2(config-ip-expl-path)# next-address 24.1.1.1
router-2(config-ip-expl-path)# next-address 25.1.1.1
router-2(config-ip-expl-path)# next-address 27.1.1.1
router-2(config-ip-expl-path)# exit
```

Note that autoroute is not used, as that could cause the Interior Gateway Protocol (IGP) to send other traffic down these tunnels.

At the inbound physical interface (in the figure above this is FE2/1), packets received are rate-limited to:

1. a rate of 30 Mbps
2. a normal burst of 1 MB
3. a maximum burst of 2 MB

Packets that are mapped to qos-group 6 and that conform to the rate-limit are marked with experimental value 5 and the BGP destination community string, and are forwarded; packets that do not conform (exceed action) are dropped:

```
router-2(config)# interface FastEthernet2/1
router-2(config-if)# rate-limit input qos-group 6 30000000 1000000 2000000 \ conform-action
 set-mpls-exp-transmit 5 exceed-action drop
router-2(config-if)# bgp-policy destination ip-qos-map
router-1(config-if)# exit
```

At the device level create a class of traffic called “exp5-class” that has MPLS experimental bit set to 5:

```
router-2(config)# class-map match-all exp5-class
router-2(config-cmap)# match mpls experimental 5
router-2(config-cmap)# exit
```

Create a policy that creates a priority queue for “exp5-class”:

```
router-2(config)# policy-map core-out-policy
router-2(config-pmap)# class exp5-class
router-2(config-pmap-c)# priority 100000
router-2(config-pmap-c)# exit
router-2(config-pmap)# class class-default
router-2(config-pmap-c)# bandwidth 55000
router-2(config-pmap-c)# exit
router-2(config-pmap)# exit
```

The policy is applied to packets exiting interface POS0/0:

```
interface POS0/0
service-policy output core-out-policy
```

As a result of all the above configuration lines, packets entering the Head-2 router via interface FE2/1 and destined for AS5, BGP community 100:1, or transiting AS5 will have their experimental field set to 5. It is assumed that no other packets entering this router (on any interface) are using this exp bit value. (If this cannot be assumed, an additional configuration must be added to mark all such packets with another experimental value.) When exiting this router via interface POS0/0, packets marked with experimental value 5 are placed into the priority queue.

Note Packets entering the router via FE2/1 and exiting through POS0/0 enter as IP packets and exit as MPLS packets.

Create a table map under BGP to map (tie) the prefixes to a qos-group. At the device level:

```
router-2(config)# ip bgp-community new-format
router-2(config)# router bgp 2
router-2(config-router)# no synchronization
router-2(config-router)# table-map set-qos-group
router-2(config-router)# bgp log-neighbor-changes
router-2(config-router)# neighbor 27.1.1.1 remote-as 2
router-2(config-router)# neighbor 27.1.1.1 update-source Loopback0
router-2(config-router)# no auto-summary
router-2(config-router)# exit
```

Create a distinct route map for this service. This includes setting the next-hop of packets matching 29.1.1.1 so they will be mapped onto Tunnel #3 (the guaranteed bandwidth service tunnel). At the device level:

```
router-2(config)# route-map set-qos-group permit 10
router-2(config-route-map)# match as-path 100
router-2(config-route-map)# set ip qos-group 6
router-2(config-route-map)# set ip next-hop 29.1.1.1
router-2(config-route-map)# exit
router-2(config)# ip as-path access-list 100 permit ^5$
```

Create a distinct route map for this service. (Its traffic will go to AS6 and AS7).

At the device level:

```
router-2(config)# route-map set-qos-group permit 10
router-2(config-route-map)# match as-path 101
router-2(config-route-map)# set ip qos-group 6
router-2(config-route-map)# set ip next-hop 29.1.1.1
router-2(config-route-map)# exit
router-2(config)# ip as-path access-list 101 permit _5_
```

Create a distinct route map for all traffic destined to prefixes that have community value 100:1. This traffic will go to AS3, AS5, and AS8.

At the device level:

```
router-2(config)# route-map set-qos-group permit 10
router-2(config-route-map)# match community 20
router-2(config-route-map)# set ip qos-group 6
router-2(config-route-map)# set ip next-hop 29.1.1.1
router-2(config-route-map)# exit
router-2(config)# ip community-list 20 permit 100:1
```

Map all guaranteed bandwidth traffic onto Tunnel #3:

```
router-2(config)# ip route 29.1.1.1 255.255.255.255 Tunnel3
```

Map all best-effort traffic onto Tunnel #4:

```
router-2(config)# ip route 30.1.1.1 255.255.255.255 Tunnel4
```

Tunnel Midpoint Configuration Mid-1 Example

All four interfaces on the midpoint router are configured very much like the outbound interface of the head router. The strategy is to have all mid-point routers in this Autonomous System ready to carry future as well as presently configured sub-pool and global pool tunnels.

At the device level:

```
router-3(config)# ip cef distributed
router-3(config)# mpls traffic-eng tunnels
```

[now one uses either the IS-IS commands on the left or the OSPF commands on the right]:

`router-3(config)# router isis`	`router ospf 100`
`router-3(config-router)# net 49.0000.2400.0000.0011.00`	`redistribute connected`
`router-3(config-router)# metric-style wide`	`network 10.1.1.0 0.0.0.255 area 0`
`router-3(config-router)# is-type level-1`	`network 11.1.1.0 0.0.0.255 area 0`
`router-3(config-router)# mpls traffic-eng level-1`	`network 24.1.1.1 0.0.0.0 area 0`
`router-3(config-router)#`	`network 12.1.1.0 0.0.0.255 area 0`
`router-3(config-router)#`	`network 13.1.1.0 0.0.0.255 area 0`
`router-3(config-router)#`	`mpls traffic-eng area 0`

[now one resumes the common command set]:

```
router-3(config-router)# mpls traffic-eng router-id Loopback0
router-3(config-router)# exit
```

Create a virtual interface:

```
router-3(config)# interface Loopback0
router-3(config-if)# ip address 24.1.1.1 255.255.255.255
router-3(config-if)# exit
```

At the physical interface level (ingress):

```
router-3(config)# interface pos2/1
router-3(config-if)# ip address 10.1.1.2 255.0.0.0
router-3(config-if)# mpls traffic-eng tunnels
router-3(config-if)# ip rsvp bandwidth 140000 140000 sub-pool 70000
[and if using IS-IS instead of OSPF]:
router-3(config-if)# ip router isis
[and in all cases]:
router-3(config-if)# exit

router-3(config)# interface pos1/1
router-3(config-if)# ip address 11.1.1.2 255.0.0.0
router-3(config-if)# mpls traffic-eng tunnels
router-3(config-if)# ip rsvp bandwidth 140000 140000 sub-pool 70000
[and if using IS-IS instead of OSPF]:
router-3(config-if)# ip router isis
[and in all cases]:
router-3(config-if)# exit
```

At the physical interface level (egress), through which two sub-pool tunnels currently exit:

```
router-3(config)# interface pos3/1
router-3(config-if)# ip address 12.1.1.1 255.0.0.0
router-3(config-if)# mpls traffic-eng tunnels
router-3(config-if)# ip rsvp bandwidth 140000 140000 sub-pool 70000
[and if using IS-IS instead of OSPF]:
router-3(config-if)# ip router isis
[and in all cases]:
router-3(config-if)# exit
```

At the physical interface level (egress), through which two global pool tunnels currently exit:

```
router-3(config)# interface pos4/1
router-3(config-if)# ip address 13.1.1.1 255.0.0.0
router-3(config-if)# mpls traffic-eng tunnels
router-3(config-if)# ip rsvp bandwidth 140000 140000 sub-pool 70000
[and if using IS-IS instead of OSPF]:
router-3(config-if)# ip router isis
[and in all cases]:
router-3(config-if)# exit
```

Tunnel Midpoint Configuration Mid-2 Example

Both interfaces on this midpoint router are configured like the outbound interfaces of the Mid-1 router.

At the device level:

```
router-5(config)# ip cef distributed
router-5(config)# mpls traffic-eng tunnels
```

[now one uses either the IS-IS commands on the left or the OSPF commands on the right]:

`router-5(config)# router isis`	`router ospf 100`
`router-5(config-router)# net 49.2500.1000.0000.0012.00`	`redistribute connected`
`router-5(config-router)# metric-style wide`	`network 13.1.1.0 0.0.0.255 area 0`
`router-5(config-router)# is-type level-1`	`network 14.1.1.0 0.0.0.255 area 0`
`router-5(config-router)# mpls traffic-eng level-1`	`network 25.1.1.1 0.0.0.0 area 0`
`router-5(config-router)#`	`mpls traffic-eng area 0`

[now one resumes the common command set]:

```
router-5(config-router)# mpls traffic-eng router-id Loopback0
router-5(config-router)# exit
```

Create a virtual interface:

```
router-5(config)# interface Loopback0
router-5(config-if)# ip address 25.1.1.1 255.255.255.255
router-5(config-if)# exit
```

At the physical interface level (ingress):

```
router-5(config)# interface pos1/1
router-5(config-if)# ip address 13.1.1.2 255.0.0.0
router-5(config-if)# mpls traffic-eng tunnels
router-5(config-if)# ip rsvp bandwidth 140000 140000 sub-pool 70000
[and if using IS-IS instead of OSPF]:
router-5(config-if)# ip router isis
[and in all cases]:
router-5(config-if)# exit
```

At the physical interface level (egress):

```
router-5(config)# interface pos2/1
router-5(config-if)# ip address 14.1.1.1 255.0.0.0
router-5(config-if)# mpls traffic-eng tunnels
router-5(config-if)# ip rsvp bandwidth 140000 140000 sub-pool 70000
[and if using IS-IS instead of OSPF]:
router-5(config-if)# ip router isis
[and in all cases]:
router-5(config-if)# exit
```

Tunnel Tail Configuration Example

The inbound interfaces on the tail router are configured much like the outbound interfaces of the midpoint routers:

At the device level:

```
router-4(config)# ip cef distributed
router-4(config)# mpls traffic-eng tunnels
```

[now one uses either the IS-IS commands on the left or the OSPF commands on the right. In the case of OSPF, one must advertise two new loopback interfaces--29.1.1.1 and 30.1.1.1 in our example--which are defined in the QoS Policy Propagation section, further along on this page]:

`router-4(config)# router isis`	`router ospf 100`
`router-4(config-router)# net 49.0000.2700.0000.0000.00`	`redistribute connected`
`router-4(config-router)# metric-style wide`	`network 12.1.1.0 0.0.0.255 area 0`
`router-4(config-router)# is-type level-1`	`network 14.1.1.0 0.0.0.255 area 0`
`router-4(config-router)# mpls traffic-eng level-1`	`network 27.1.1.1 0.0.0.0 area 0`
`router-4(config-router)#`	`network 29.1.1.1 0.0.0.0 area 0`
`router-4(config-router)#`	`network 30.1.1.1 0.0.0.0 area 0`
`router-4(config-router)#`	`mpls traffic-eng area 0`

[now one resumes the common command set, taking care to include the two additional loopback interfaces]:

```
router-4(config-router)# mpls traffic-eng router-id Loopback0
router-4(config-router)# mpls traffic-eng router-id Loopback1
router-4(config-router)# mpls traffic-eng router-id Loopback2
router-4(config-router)# exit
```

Create a virtual interface:

```
router-4(config)# interface Loopback0
router-4(config-if)# ip address 27.1.1.1 255.255.255.255
router-4(config-if)# exit
```

At the physical interface (ingress):

```
router-4(config)# interface pos2/1
router-4(config-if)# ip address 12.1.1.2 255.0.0.0
router-4(config-if)# mpls traffic-eng tunnels
router-4(config-if)# ip rsvp bandwidth 140000 140000 sub-pool 70000
[and if using IS-IS instead of OSPF]:
router-4(config-if)# ip router isis
[and in all cases]:
router-4(config-if)# exit
router-4(config)# interface pos2/2
router-4(config-if)# ip address 14.1.1.2 255.0.0.0
router-4(config-if)# mpls traffic-eng tunnels
router-4(config-if)# ip rsvp bandwidth 140000 140000 sub-pool 70000
```

```
[and if using IS-IS instead of OSPF]:
router-4(config-if)# ip router isis
[and in all cases]:
router-4(config-if)# exit
```

On the tail device, one must configure a separate virtual loopback IP address for each class-of-service terminating here. The headend routers need these addresses to map traffic into the proper tunnels. In the current example, four tunnels terminate on the same tail device but they represent only two service classes, so only two additional loopback addresses are needed:

Create two virtual interfaces:

```
router-4(config)# interface Loopback1
router-4(config-if)# ip address 29.1.1.1 255.255.255.255
[and if using IS-IS instead of OSPF]:
router-4(config-if)# ip router isis
[and in all cases]:
router-4(config-if)# exit
router-4(config)# interface Loopback2
router-4(config-if)# ip address 30.1.1.1 255.255.255.255
[and if using IS-IS instead of OSPF]:
router-4(config-if)# ip router isis
[and in all cases]:
router-4(config-if)# exit
```

At the device level, configure BGP to send the community to each tunnel head:

```
router-4(config)# ip bgp-community new-format
router-4(config)# router bgp 2
router-4(config-router)# neighbor 23.1.1.1 send-community
router-4(config-router)# neighbor 22.1.1.1 send-community
router-4(config-router)# exit
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
IS-IS	Cisco IOS IP Routing: ISIS Command Reference
MPLS commands	Cisco IOS MPLS Command Reference
OSPF	Cisco IOS IP Routing: OSPF Command Reference
QoS	Cisco IOS Quality of Service Solutions Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 3270	*Requirements for Support of Diff-Serv-aware MPLS Traffic Engineering* Multi-Protocol Label Switching (MPLS) Support of Differentiated Services F. Le Faucheur, L. Wu, B. Davie, P. Vaananen, R. Krishnan, P. Cheval, & J. Heinanen
RFC 4124	Protocol Extensions for Support of Diffserv-aware MPLS Traffic Engineering ed. by F. Le Faucheur
RFC 4127	Russian Dolls Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering ed. by F. Le Faucheur
RFC 4125	Maximum Allocation Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering by F. Le Faucheur & W. Lai
IETF Diff-Serv-aware MPLS Traffic Engineering	The new concept of "Class-Type" defined in the IETF Standard corresponds to the prior concept of “bandwidth pool” that was implemented in the original version of DS-TE. Likewise, the two bandwidth pools implemented in the original version of DS-TE (global pool and sub-pool) correspond to two of the IETF Standard’s new Class-Types (Class-Type 0 and Class-Type 1, respectively).

MIBs

MIB	MIBs Link
None	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/cisco/web/support/index.html

Glossary

This section defines acronyms and words that may not be readily understood.

AS --Autonomous System. A collection of networks under a common administration, sharing a common routing strategy and identified by a unique 16-bit number (assigned by the Internet Assigned Numbers Authority).

BGP --Border Gateway Protocol. The predominant interdomain routing protocol. It is defined by RFC 1163. Version 4 uses route aggregation mechanisms to reduce the size of routing tables.

CBR --Constraint Based Routing. The computation of traffic paths that simultaneously satisfy label-switched path attributes and current network resource limitations.

CEF --Cisco Express Forwarding. A means for accelerating the forwarding of packets within a router, by storing route lookup information in several data structures instead of in a route cache.

CLI --Command Line Interface. Cisco's interface for configuring and managing its routers.

DS-TE --Diff Serv-aware Traffic Engineering. The capability to configure two bandwidth pools on each link, a *global pool* and a *sub-pool*. MPLS traffic engineering tunnels using the sub-pool bandwidth can be configured with Quality of Service mechanisms to deliver guaranteed bandwidth services end-to-end across the network. Simultaneously, tunnels using the global pool can convey DiffServ traffic.

flooding --A traffic passing technique used by switches and bridges in which traffic received on an interface is sent out through all of the interfaces of that device except the interface on which the information was originally received.

GB queue --Guaranteed Bandwidth queue. A per-hop behavior (PHB) used exclusively by the strict guarantee traffic. If delay/jitter guarantees are sought, the diffserv Expedited Forwarding queue (EF PHB) is used. If only bandwidth guarantees are sought, the diffserv Assured Forwarding PHB (AF PHB) is used.

Global Pool --The total bandwidth allocated to an MPLS traffic engineering link.

IGP --Interior Gateway Protocol. An internet protocol used to exchange routing information within an autonomous system. Examples of common internet IGPs include IGRP, OSPF, and RIP.

label-switched path (LSP) tunnel --A configured connection between two routers, using label switching to carry the packets.

IS-IS --Intermediate System-to-Intermediate System. A link-state hierarchical routing protocol, based on DECnet Phase V routing, whereby nodes exchange routing information based on a single metric, to determine network topology.

LCAC --Link-level (per-hop) call admission control.

LSP --Label-switched path (see above). Also Link-state packet--A broadcast packet used by link-state protocols that contains information about neighbors and path costs. LSPs are used by the receiving routers to maintain their routing tables. Also called link-state advertisement (LSA).

MPLS --Multi-Protocol Label Switching (formerly known as Tag Switching). A method for directing packets primarily through Layer 2 switching rather than Layer 3 routing, by assigning the packets short fixed-length labels at the ingress to an MPLS cloud, using the concept of forwarding equivalence classes. Within the MPLS domain, the labels are used to make forwarding decisions mostly without recourse to the original packet headers.

MPLS TE --MPLS Traffic Engineering (formerly known as "RRR" or Resource Reservation Routing). The use of label switching to improve traffic performance along with an efficient use of network resources.

OSPF --Open Shortest Path First. A link-state, hierarchical IGP routing algorithm, derived from the IS-IS protocol. OSPF features include least-cost routing, multipath routing, and load balancing.

RSVP --Resource reSerVation Protocol. An IETF protocol used for signaling requests (to set aside internet services) by a customer before that customer is permitted to transmit data over that portion of the network.

Sub-pool --The more restrictive bandwidth in an MPLS traffic engineering link. The sub-pool is a portion of the link's overall global pool bandwidth.

TE --Traffic engineering. The application of scientific principles and technology to measure, model, and control internet traffic in order to simultaneously optimize traffic performance and network resource utilization.

Feature Information for MPLS Traffic Engineering - DiffServ Aware (DS-TE)

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.

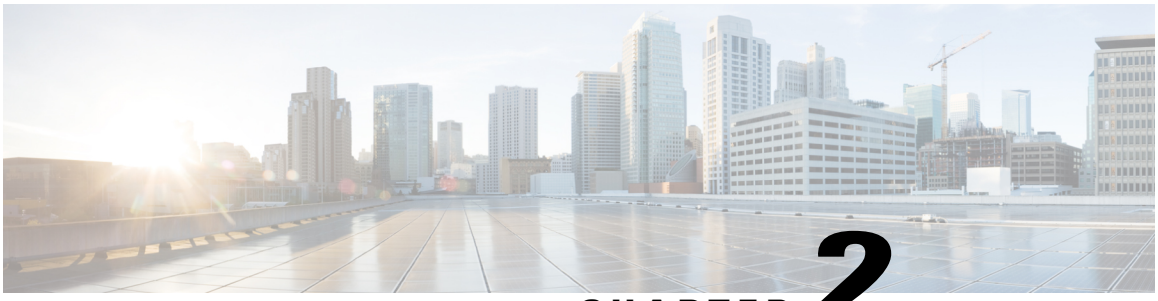
Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 3: Feature Information for MPLS Traffic Engineering - DiffServe Aware (DS-TE)

Feature Name	Releases	Feature Information
MPLS Traffic Engineering - DiffServ Aware (DS-TE)	12.0(11) ST 12.0(14) ST 12.0(14) ST-1 12.0(22)S 12.2(14)S 12.2(18)S 12.2(18)SXD 12.2(28)SB 12.2(33)SRB Cisco IOS XE Release 3.5S	

Feature Name	Releases	Feature Information
		The Multiprotocol Label Switching Traffic Engineering (MPLS TE) - DiffServ-Aware Traffic Engineering (DS-TE) feature enables service providers to perform separate admission control and separate route computation for discrete subsets of traffic (for example, voice and data traffic). When DS-TE is combined with other Cisco software features such as QoS, the service provider can: • Develop QoS services for end customers based on *signaled* rather than *provisioned* QoS • Build the higher-revenue generating "strict-commitment" QoS services, without over-provisioning • Offer virtual IP leased-line, Layer 2 service emulation, and point-to-point guaranteed bandwidth services including voice-trunking • Enjoy the scalability properties offered by MPLS. DS-TE feature introduced in Cisco IOS Release 12.0(11) ST. In Cisco IOS Release 12.0(14) ST-1, support was added for guaranteed bandwidth service directed to many destination prefixes (for example, guaranteed bandwidth service destined to an autonomous system or to a BGP community). In Cisco IOS Release 12.0(14) ST-1, support was added for the IS-IS Interior Gateway Protocol. In Cisco IOS Release 12.0(14) ST, support was added for the Cisco Series 7500(VIP) platform. Feature was implemented in Cisco

Feature Name	Releases	Feature Information
		IOS Release 12.0(22)S. Feature was integrated into Cisco IOS Release 12.2(14)S. Feature was implemented in Cisco IOS Release 12.2(18)S. Feature was implemented in Cisco IOS Release 12.2(18)SXD. Feature was implemented in Cisco IOS Release 12.2(28)SB. In Cisco IOS Release 12.2(33)SRB, this feature was augmented to include the new IETF-Standard functionality of DS-TE, as described in RFCs 3270, 4124, 4125, and 4127. Feature was implemented in Cisco IOS XE Release 3.5S.



CHAPTER 2

MPLS DiffServ Tunneling Modes

MPLS DiffServ Tunneling Modes allows service providers to manage the quality of service (QoS) that a router will provide to a Multiprotocol Label Switching (MPLS) packet in an MPLS network. MPLS DiffServ Tunneling Modes conforms to the IETF draft standard for Uniform, Short Pipe, and Pipe modes. It also conforms to Cisco-defined extensions for scalable command line interface (CLI) management of those modes at customer edge, provider edge, and core routers.

The following features are supported on MPLS DiffServ Tunneling Modes:

- MPLS per-hop behavior (PHB) layer management.
- There is improved scalability of the MPLS layer management by control on managed customer edge (CE) routers.
- MPLS can “tunnel” a packet’s QoS (that is, the QoS is transparent from edge to edge).
- The MPLS experimental (MPLS EXP) field can be marked differently and independently of the PHB marked in the IP Precedence or differentiated services code point (DSCP) field.
- There are three MPLS QoS tunneling modes for the operation and interaction between the DiffServ marking in the IP header and the DiffServ marking in the MPLS header: Pipe mode with an explicit NULL LSP, Short Pipe mode, and Uniform mode. Pipe mode with an explicit NULL LSP and Short Pipe mode allow an MPLS network to transparently tunnel the DiffServ marking of packets.

MPLS DiffServ Tunneling Modes has the following benefits:

- Tunneling modes provide added QoS functionality by the creative manipulation of the MPLS EXP field during label imposition, forwarding, and label disposition.
- Tunneling modes provide a common set of PHBs to different service provider customers.
- Pipe mode provides transparency and customized edge service.
- Pipe mode with an explicit NULL LSP improves the scalability of management by performing per-customer packet metering and marking closer to the service provider’s customer networks.
- Pipe mode with an explicit NULL LSP provides QoS transparency by ensuring that customer’s packets will not be re-marked in the service provider’s network.
- In Pipe mode with an explicit NULL LSP, the explicit NULL LSP applies the service provider’s PHBs on the ingress CE-to-PE link.

- In Pipe mode with an explicit NULL LSP, the service provider's PHBs are applied on the egress PE-to-CE link.
- Short Pipe mode provides transparency, standard edge service, and scalability.
- Short Pipe mode provides PHB management on the PE router. The customer's set of PHBs is applied on both the egress PE-to-CE link and on the ingress CE-to-PE link.
- Customers are likely to use Uniform mode if they have no markings or few markings. The customer lets the Internet service provider (ISP) mark the packets and retain their markings.
- In Uniform mode, all changes to QoS markings are reflected at each level (that is, IGP, BGP, and IP).
- In Uniform mode, if a QoS marking is changed in the MPLS network, it is changed in the IP packet too.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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 table at the end of this module.

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.

Prerequisites for MPLS DiffServ Tunneling Modes

- Set up the network to run MPLS.
- Enable IP Cisco Express Forwarding (CEF).
- Define the Service Level Agreement (SLA).
- Know each customer's per-hop behavior.
    - What do customers expect you to provide?
    - Are customers going to mark the traffic?

- Identify whether the customer's traffic will be voice or data.
- Determine the topology and interfaces that need to be configured.
- Understand how IP and MPLS packets are forwarded.

Restrictions for MPLS DiffServ Tunneling Modes

- A single label-switched path (LSP) can support up to eight classes of traffic (that is, eight PHBs) because the MPLS EXP field is a 3-bit field.
- MPLS DiffServ Tunneling Modes does not support L-LSPs. Only E-LSPs are supported.

Information About MPLS DiffServ Tunneling Modes

QoS and Its Use in MPLS Tunneling

This section includes the following subsections:

What is QoS

Critical applications must be guaranteed the network resources they need, despite a varying network traffic load. QoS is a set of techniques that manage the following:

- Network bandwidth--Noncritical traffic is prevented from using bandwidth that critical applications need. The main cause of congestion is lack of bandwidth.
- Network delay (also called latency)--The time required to move a packet from the source to the destination over a path.
- Jitter--The interpacket delay variance; that is, the difference between interpacket arrival and departure. Jitter can cause data loss.
- Packet loss--The dropping of packets.

Service providers offering MPLS VPN and traffic engineering (TE) services can provide varying levels of QoS for different types of network traffic. For example, Voice-over-IP (VoIP) traffic receives service with an assured minimum of delay, whereas e-commerce traffic might receive a minimum bandwidth guarantee (but not a delay guarantee).

For more information about QoS, see the *Cisco IOS Quality of Service Solutions Configuration Guide* , Release 12.2 and the *Cisco IOS Quality of Service Solutions Command Reference* , Release 12.2.

Services Supported by MPLS QoS

MPLS QoS supports the following services:

- Class-based weighted fair queuing (CBWFQ)--Provides queuing based on defined classes, with no strict priority queue available for real-time traffic. Weighted fair queuing allows you to define traffic classes

based on match criteria. Once a class has been defined, you can assign characteristics to the class. For example, you can designate the minimum bandwidth delivered to the class during congestion.

- Low latency queuing (LLQ)--Provides strict priority queuing, which allows delay-sensitive data such as voice to be processed and sent first, before packets in other queues are processed. This provides preferential treatment to delay-sensitive data over other traffic.
- Weighed fair queuing (WFQ)--An automated scheduling system that uses a queuing algorithm to ensure fair bandwidth allocation to all network traffic. Weighted fair queuing is based on a relative bandwidth applied to each of the queues.
- Weighted random early detection (WRED)--RED is a congestion avoidance mechanism that controls the average queue size by indicating to the end hosts when they should temporarily stop sending packets. A small percentage of packets is dropped when congestion is detected and before the queue in question overflows completely.

The weighted aspect of WRED ensures that high-precedence traffic has lower loss rates than other traffic during congestion. WRED can be configured to discard packets that have certain markings. When a packet comes into a router, it is assigned an internal variable that is called a discard class. If desired, you can set the discard class at the input interface. At the output interface, the router can be configured to use the discard class for WRED instead of the MPLS EXP field.

Service Level Agreements Used in MPLS Tunneling

The service provider has an SLA with each customer. Each customer can have a different SLA. For example, the SLA for customer C1 may allow 256 kilobits of bandwidth for TCP packets (such as FTP packets or Telnet packets) and 1 megabyte of voice traffic per second. If the customer transmits 1 megabyte of voice traffic per second, the service provider delivers it to the other side of the customer's network. If the customer transmits more, the excess traffic is considered out-of-rate traffic and may or may not be discarded.

If the service provider experiences congestion, the service provider decides how to handle those packets. For example, the service provider may drop packets or give them less bandwidth. The PHB may be to drop a packet or to give it 20 percent of the link bandwidth.

The PHB that the service provider provides for a packet may be different from the PHB that the customer wants traffic to have in their network. The customer may be providing QoS at the output interface of each router in their network. However, the customer may be providing a different amount of bandwidth on those links than the service provider will provide. For example, a customer may give 50 percent of the link bandwidth to voice. The service provider may want to give only 10 percent of the link bandwidth to voice.

Providing QoS to an IP Packet

In an IP packet, the QoS that a router must provide has traditionally been designated in the IP Precedence field, which is the first three bits of the type of service (ToS) byte in the header of an IP packet. The IP Precedence and the differentiated services code point (DSCP) in an IP packet define the class. They may also designate the discard profile within a class. The DSCP is specified in the IETF standard for DiffServ. It is a new IETF standard for QoS.

Although some people still use the IP Precedence field, others use the DSCP to indicate the PHB that will be provided to an IP packet.

After label imposition, a configurable mapping function marks an equivalent PHB into the 3-bit MPLS EXP field value based on the IP Precedence or the IP DSCP marking.

Providing QoS to an MPLS Packet

In an MPLS packet, the PHB is marked in the MPLS EXP field within the MPLS label entry.

The EXP bits are similar in function to the IP Precedence and the DSCP in the IP network. The EXP bits generally carry all the information encoded in the IP Precedence or the DSCP.

The edge LSR that imposes the MPLS header sets the MPLS EXP field to a value.

DiffServ as a Standardization of QoS

DiffServ is a QoS architecture for IP networks. Packets within a DiffServ-enabled network may be classified into classes such as premium, gold, silver, or bronze based on QoS requirements. For example, VoIP packets may be grouped into the premium class, and e-commerce HTTP packets may be grouped into the gold class.

Each class has a marking associated with it. This makes packet classification extremely scalable and assures appropriate bandwidth and delay guarantees in the network. When packets enter the network, they are marked based on classification policies at the network boundary routers. The boundary routers also apply traffic conditioning functions to control the amount of traffic entering the network. Traffic conditioning includes the following:

- Shaping--Smoothing the rate at which packets are sent into the network
- Policing--Dropping packets that exceed a subscribed-to-rate, or re-marking packets exceeding the rate so that the probability of dropping them increases when there is congestion

Each router within the network then applies different queuing and dropping policies on each packet based on the marking that the packet carries.

For more information about DiffServ, see the *Cisco IOS Switching Services Configuration Guide* , Release 2.2.

Tunneling Modes for MPLS DiffServ

Tunneling is the ability of QoS to be transparent from one edge of a network to the other edge of the network. A tunnel starts where there is label imposition. A tunnel ends where there is label disposition; that is, where the label is popped off of the stack and the packet goes out as an MPLS packet with a different PHB layer underneath or as an IP packet with the IP PHB layer.

There are three ways to forward packets through a network:

- Pipe mode with an explicit NULL LSP
- Short Pipe mode
- Uniform mode

Pipe mode and Short Pipe mode provide QoS transparency. With QoS transparency, the customer's IP marking in the IP packet is preserved.

Note The only difference between Pipe mode and Short Pipe mode is which PHB is used on the service provider's egress edge router. In Pipe mode with an explicit NULL LSP, QoS is done on the PE-to-CE link based on the service provider's PHB markings. The egress LSR still uses the marking that was used by intermediate LSRs.

All three tunneling modes affect the behavior of edge and penultimate label switching routers (LSRs) where labels are pushed (put onto packets) and popped (removed from packets). They do not affect label swapping at intermediate routers. A service provider can choose different types of tunneling modes for each customer.

Following is a brief description of each tunneling mode:

- Pipe mode with an explicit NULL LSP--QoS is done on the output interface of the PE router based on the received MPLS EXP field, even though one or more label entries have been popped. The IP Precedence field, EXP bits, and the DSCP field are not altered when they travel from the ingress to the egress of the MPLS network.

Any changes to the packet marking within the MPLS network are not permanent and do not get propagated when the packet leaves the MPLS network. The egress LSR still uses the marking that was used by intermediate provider core (P) routers. However, the egress provider edge (PE) router has to remove labels imposed on the original packet. To preserve the marking carried in the labels, the edge PE router keeps an internal copy of the marking before removing the labels. This internal copy is used to classify the packet on the outbound interface (facing the CE) after the labels are removed.

For a detailed description, see the Pipe Mode with an Explicit NULL LSP, on page 55.

For the configuration procedure, see the Configuring Pipe Mode with an Explicit NULL LSP, on page 64.

For an example, see the Pipe Mode with an Explicit NULL LSP Configuration Example, on page 90.

- Short Pipe mode--In Short Pipe mode, the egress PE router uses the original packet marking instead of the marking used by the intermediate P routers.

For a detailed description, see the Short Pipe Mode, on page 59.

For the configuration procedure, see the Configuring Short Pipe Mode, on page 74.

For an example, see the Short Pipe Mode Configuration Example, on page 92.

- Uniform mode--In Uniform mode, the marking in the IP packet may be manipulated to reflect the service provider's QoS marking in the core.

For a detailed description, see the Uniform Mode, on page 62.

For the configuration procedure, see the Configuring Uniform Mode, on page 80.

For an example, see the Uniform Mode Configuration Example, on page 93.

MPLS PHB Layer Management

Through the network of routers, the MPLS EXP field can be marked differently and independently of the PHB marked in the IP Precedence or the DSCP field. A service provider can choose from existing classification criteria, including or excluding the IP PHB marking, to classify packets into a different PHB which is then marked only in the MPLS EXP field during label imposition.

Layer management is the ability to apply an additional layer of PHB marking to a packet. The PHB is the behavior of a packet at a router (that is, the unique discard and scheduling behavior that is applied to a packet). Layer management can occur at a service provider-managed CE router or at the service provider edge (PE) router.

If a packet arrives in a network as an IP packet, it may already have a PHB layer that is represented by a marking in the ToS byte. The marking can be IP Precedence bits or the DSCP.

If a packet arrives as an MPLS packet, it already has the following two PHB layers:

- IP layer
- MPLS layer, where the marking is in the MPLS EXP field of the topmost label entry

At a given hop, one PHB layer can be added to a packet. If only one label is being pushed onto the packet, the marking for the PHB layer being added is contained in only one label.

If two or more labels are being pushed onto a packet, the PHB layer being added is marked with the same MPLS EXP field in all of the label entries being pushed on at that time.

Tunneling Modes Operation

Note Cisco IOS allows a flexible configuration. You can configure the PHB definition of the MPLS EXP field differently from the PHB definition of the IP Precedence and DSCP.

A service provider may or may not care about the PHB marking of their customer's packet. For example, in customer C1's network, an IP Precedence value of 5 may mean voice. In customer C2's network, an IP Precedence value of 3 may mean voice. The service provider does not want to have two different IP Precedence values for voice. If the service provider has a large number of customers, there could be "many" values for voice. There are only eight possible EXP values.

To deal with different IP Precedence values representing the same PHB (in our example, for voice), the service provider does the following:

1. Arbitrarily chooses a common MPLS EXP field value to represent a PHB. For example, 2 can represent voice.
2. Looks at the packets of each customer. The service provider may look at the IP Precedence field value or at the UDP port number for voice, which is constant in every network.
3. For all customers, sets each voice packet to the MPLS EXP field value 2 on all the service provider's customer ports. Consequently, each router in the service provider's network only has to look for the MPLS EXP field value 2 for voice.

Another solution would be to set the DSCP value to 2, but that would alter the customer's PHB. MPLS DiffServ tunneling modes achieve the same results without altering the DSCP value.

This section illustrates and describes the following:

Pipe Mode with an Explicit NULL LSP

This section describes the following:

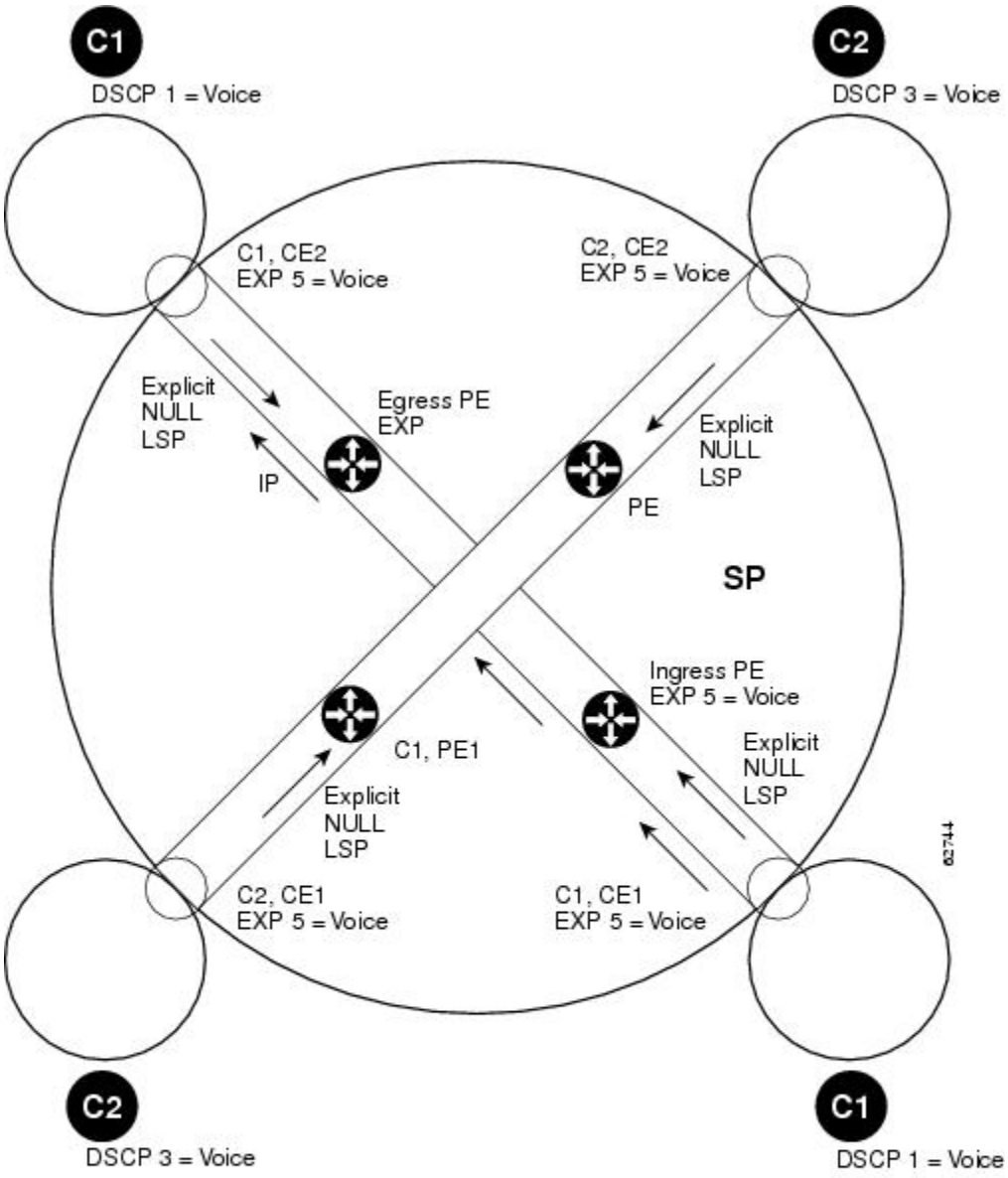
Pipe Mode with an Explicit NULL LSP Overview

Pipe mode with an explicit NULL LSP has the following characteristics:

- The QoS tunnel goes from the ingress CE router through the PE router to the egress CE router.
- There is an explicit NULL LSP from the CE router to the PE router. The label entry contains an MPLS EXP field, but does not carry a label value for forwarding purposes. It contains a zero (a null label value) for all packets going to the ingress PE router.
- The egress PE router removes the label entry and forwards packets as IP, but QoS is done on the output interface based on the MPLS EXP field received by the egress PE router.
- The service provider does not overwrite the IP Precedence value in the service provider's network.

The figure below shows an overview of Pipe mode with an explicit NULL LSP.

Figure 1: Pipe Mode with an Explicit NULL LSP Overview



Symbol	Meaning
C1	Customer 1's DiffServ domain
C2	Customer 2's DiffServ domain
CE1	Customer edge router 1
PE1	Service provider edge router (ingress LSR)
P1	Service provider router within the core of the provider's network
P2	Service provider router within the core of the provider's network
PE2	Service provider's edge router (egress LSR)
CE2	Customer edge router 2
SP	Service provider DiffServ domain

Note PE1 and PE2 are at the boundaries between the MPLS network and the IP network.

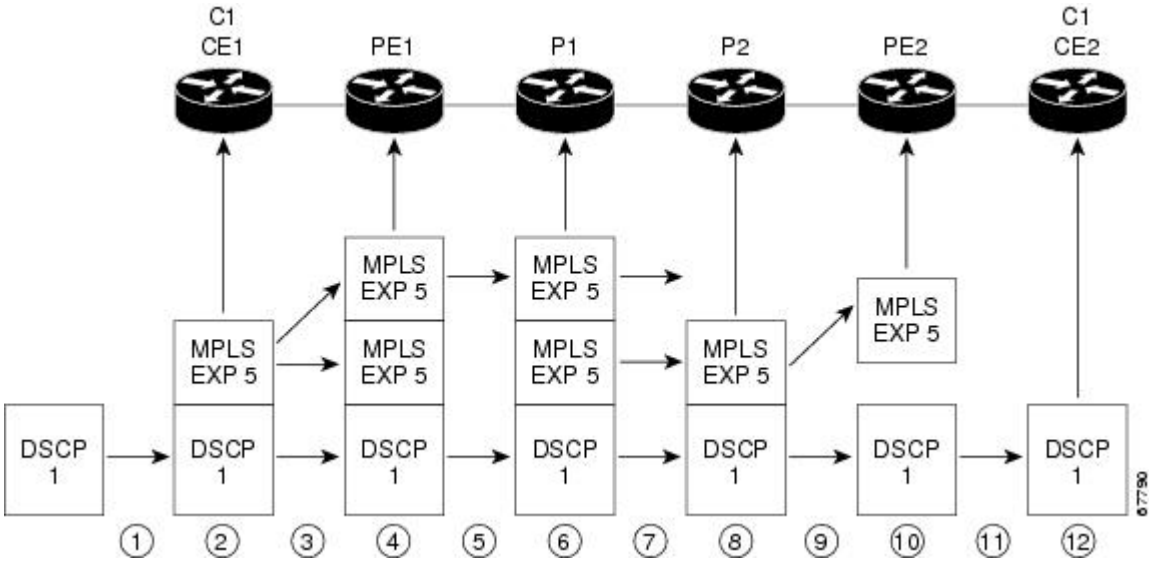
The figure above illustrates the following:

1. An IP packet arrives at C1, CE1 with a DSCP value of 1.
2. C2, CE1 sets the MPLS EXP field value to 5 during label imposition of the null label.
3. The packet goes through the service provider's network with the MPLS EXP field value set to 5.
4. Each router in the service provider's network looks at the MPLS EXP field and does QoS based on that value.
5. When the packet gets to the egress PE router going back into C1's network, it does QoS based on the packet's MPLS EXP field even though the packet is transmitted as an IP packet.

Pipe Mode with an Explicit NULL LSP Operating Procedure

The figure below illustrates the operation of Pipe mode with an explicit NULL LSP for Customer 1, when MPLS VPN is enabled. Since VPN is enabled, there are two MPLS label entries. Otherwise, there would be only one entry. The functionality would be similar for Customer 2, but the DSCP value would be 3.

Figure 2: Pipe Mode with an Explicit NULL LSP Operation with MPLS VPN Enabled



Pipe mode with an explicit NULL LSP functions as follows. The circled numbers at the bottom of the illustration correspond to the step numbers.

1. IP packets arrive at the router CE1, the managed CE router, with a DSCP value of 1.
2. An explicit NULL label entry is imposed onto the packet that contains an EXP value of 5.
3. The packet is transmitted to PE1 on the explicit NULL LSP.

4. The PE1 router saves the value of the MPLS EXP field and removes the explicit NULL entry. The PE1 router then imposes new labels onto the IP packet. Each label entry is set to the saved MPLS EXP field 5.
5. The packet is transmitted to P1.
6. At P1, the received EXP value is copied into the swapped label entry.
7. The packet is transmitted to P2.
8. At P2, the topmost label is popped, exposing a label entry that also has an EXP value of 5.
9. The packet is transmitted to PE2.
10. PE2 stores the value of the MPLS EXP field in the qos-group and discard-class variables, and removes the label entry from the packet.
11. While transmitting the packet to CE2, PE2 does QoS on its egress interface based on the saved value of the MPLS EXP field (qos-group and discard-class).
12. The IP packet arrives at the CE2 router.

Short Pipe Mode

This section describes the following:

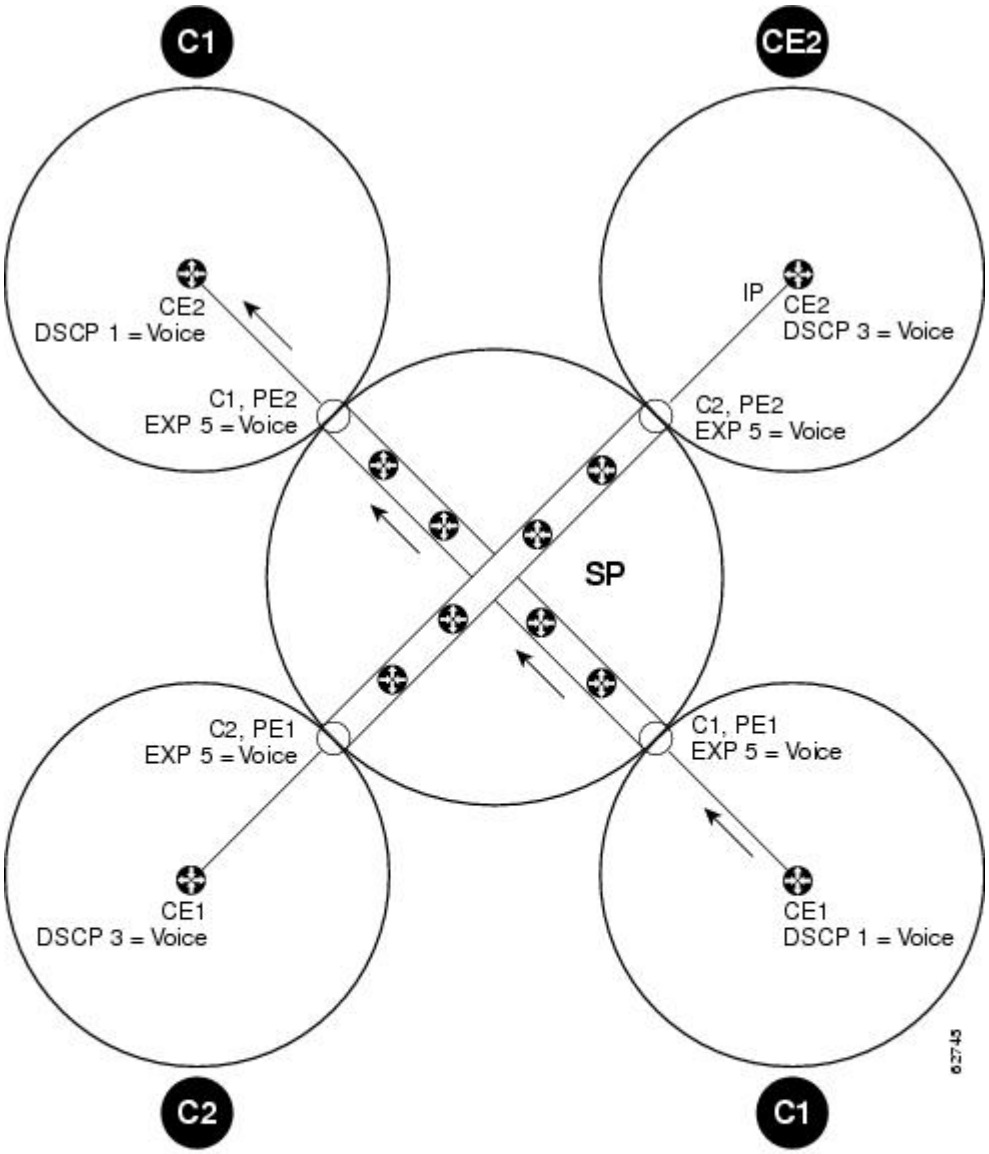
Short Pipe Mode Overview

Short Pipe mode has the following characteristics:

- The QoS tunnel goes from the ingress PE router to the egress PE router.
- The egress PE router transmits packets as IP and QoS is done on the output interface based on the IP DSCP or IP Precedence value.
- The service provider does not overwrite the DSCP or IP Precedence value in the service provider's network.

The figure below shows an overview of Short Pipe mode.

Figure 3: Short Pipe Mode Overview



Symbol	Meaning
C1	Customer 1's DiffServ domain
C2	Customer 2's DiffServ domain
CE1	Customer edge router 1
PE1	Service provider edge router (ingress LSR)
P1	Service provider router within the core of the provider's network
P2	Service provider router within the core of the provider's network
PE2	Service provider's edge router (egress LSR)
CE2	Customer edge router 2
SP	Service provider DiffServ domain

Note PE1 and PE2 are at the boundaries between the MPLS network and the IP network.

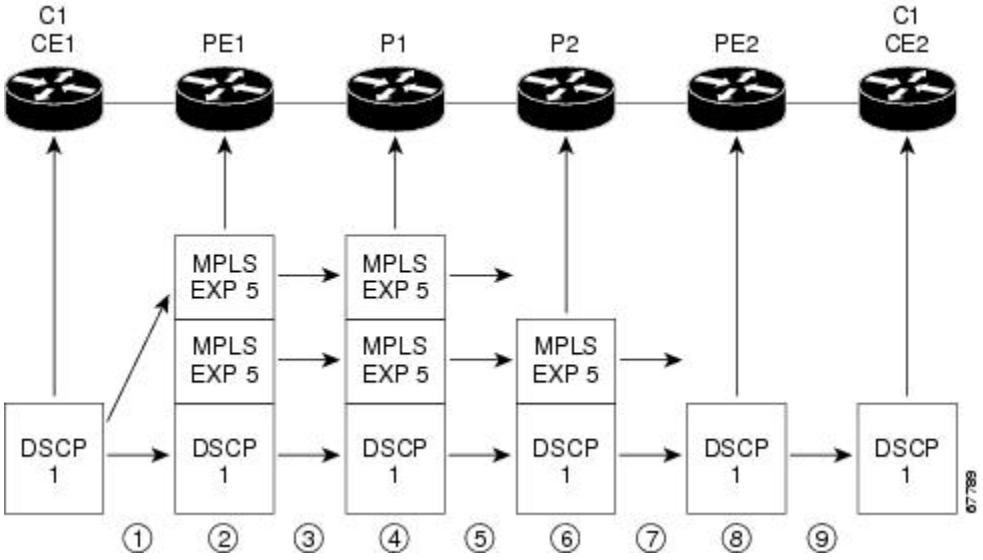
The figure above shows the following:

1. An IP packet arrives at C1, CE1 with a DSCP value of 1.
2. C1, CE1 transmits the IP packet to C1, PE1.
3. C1, PE1 sets the MPLS EXP field value to 5 during label imposition of the VPN label entries.
4. The packet goes through the service provider's network with the MPLS EXP field value set to 5.
5. Each router in the service provider's network looks at the MPLS EXP field and does QoS based on that value.
6. When the packet gets to the egress PE router going back into C1's network, it does QoS based on the IP DSCP field.

Short Pipe Mode Operating Procedure

The figure below illustrates Short Pipe mode.

Figure 4: Short Pipe Mode Operation



Short Pipe mode functions as follows. The circled numbers at the bottom of the illustration correspond to the step numbers.

1. C1, CE1 transmits an IP packet to PE1 with an IP DSCP value of 1.
2. PE1 sets the MPLS EXP field to 5 in the imposed label entries.
3. PE1 transmits the packet to P1.
4. P1 sets the MPLS EXP field value to 5 in the swapped label entry.
5. P1 transmits the packet to P2.

6 P2 pops the IGP label entry.

7 P2 transmits the packet to PE2.

8 PE2 pops the BGP label.

9 PE2 transmits the packet to C1, CE2, but does QoS based on the IP DSCP value.

Uniform Mode

This section describes the following:

Uniform Mode Overview

In a label, the MPLS EXP field is not the same as the label value.

The topmost label entry contains the following:

- Label value, which contains labels and other information, to forward the packet.
- MPLS EXP field, which only pertains to the QoS of the packet, not the route. The EXP field value is not advertised. Its value comes from the way that the packet is received.

In Uniform mode, packets are treated uniformly in the IP and MPLS networks; that is, the IP Precedence value and the MPLS EXP bits always are identical. Whenever a router changes or recolors the PHB of a packet, that change must be propagated to all encapsulation markings. The propagation is performed by a router only when a PHB is added or exposed due to label imposition or disposition on any router in the packet's path. The color must be reflected everywhere, at all levels. For example, if a packet's QoS marking is changed in the MPLS network, the IP QoS marking reflects that change.

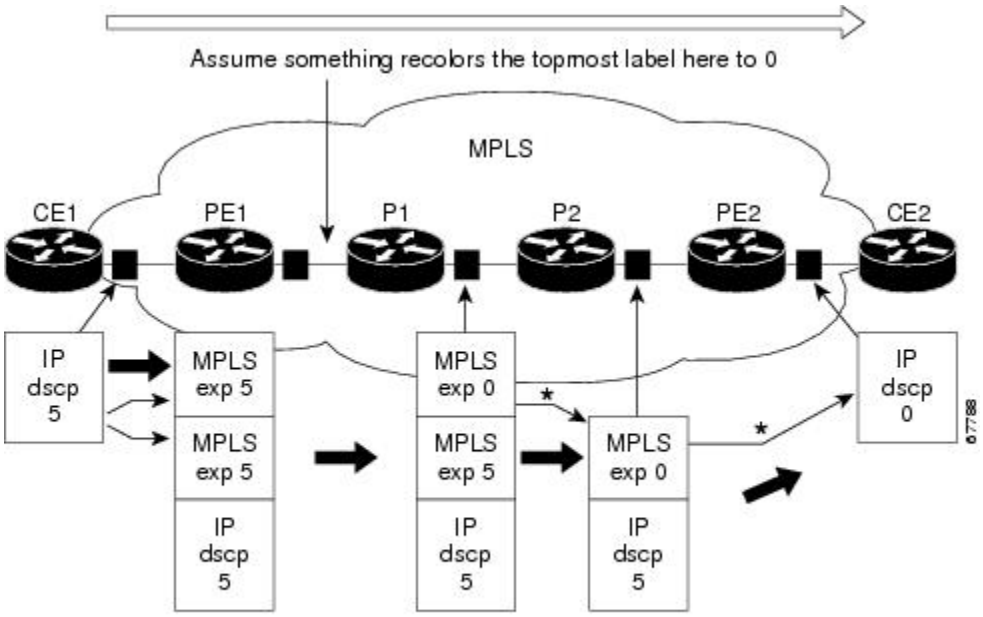
Uniform mode functions as follows:

- In both the MPLS-to-MPLS path and the MPLS-to-IP path, the PHBs of the topmost popped label are copied into the new top label or into the IP DSCP if no label remains.
- There can be a maximum of eight PHBs.
- If the PHBs are enclosed using more than the three Precedence bits, you must map DSCP to MPLS at the entry to the MPLS cloud.
- When packets leave the MPLS cloud, you must remap from the MPLS EXP value to the DSCP field in the IP header.

Uniform Mode Operating Procedure

The figure below illustrates the operation of Uniform mode.

Figure 5: Uniform Mode Operation



*In both the MPLS-to-MPLS and the MPLS-to-IP cases, the PHBs of the topmost popped label is copied into the new top label or the IP DSCP if no label remains

The procedure varies according to whether there are IP Precedence bit markings or DSCP markings.

The following actions occur if there are IP Precedence bit markings:

1. IP packets arrive in the MPLS network at PE1, the service provider edge router.
2. A label is copied onto the packet.
3. If the MPLS EXP field value is recolored (for example, if the packet becomes out-of-rate because too many packets are being transmitted), that value is copied to the IGP label. The value of the BGP label is not changed.
4. At the penultimate hop, the IGP label is removed. That value is copied into the next lower level label.
5. When all MPLS labels have been removed from the packet which is sent out as an IP packet, the IP Precedence or DSCP value is set to the last changed EXP value in the core.

Following is an example when there are IP precedence bit markings:

1. At CE1 (customer equipment 1), the IP packet has an IP Precedence value of 5.
2. When the packet arrives in the MPLS network at PE1 (the service provider edge router), the IP Precedence value of 5 is copied to the imposed label entries of the packet.
3. The MPLS EXP field in the IGP label header might be changed within the MPLS core (for example, at P1).

Note Since the IP Precedence bits are 5, the BGP label and the IGP label also contain 5 because in Uniform mode the labels always are identical. The packet is treated uniformly in the IP and MPLS networks.

1. At P2, when the IGP label is removed, the MPLS EXP field in this label entry is copied into the underlaying BGP label.
2. At PE2, when the BGP label is popped, the EXP field in this label header is copied into the IP Precedence field of the underlying IP header.

How to Configure MPLS DiffServ Tunneling Modes

Note You can configure only one of the tunneling modes.

Determining Which Tunneling Mode is Appropriate

- If there are managed customer edge (CE) routers, we recommend that you use Pipe mode with an explicit NULL LSP so that there is service provider PHB on the PE-to-CE link.
- If there is no managed CE router, we recommend that you use Short Pipe mode.
- If there are no markings or few markings, customers are likely to use Uniform mode.

Setting the MPLS EXP field

There are two ways to set the MPLS EXP field:

- Use the set mpls experimental topmost command to set the topmost label entry's value directly in the packet on the input and/or output interfaces.
- Use the set mpls experimental imposition command on the input interface to set the pushed label entry's value during label imposition.

By default, the label edge router copies the IP Precedence of the IP packet to the MPLS EXP field in all pushed label entries.

You can optionally map the IP Precedence or DSCP field to the MPLS EXP field in the MPLS header by using the **set mpls experimental imposition** command.

Configuring Pipe Mode with an Explicit NULL LSP

This section describes how to configure the following:

For examples, see the Pipe Mode with an Explicit NULL LSP Configuration Example, on page 90.

Note The steps that follow show one way, but not the only way, to configure Pipe Mode with an Explicit NULL LSP.

Ingress CE Router--Customer Facing Interface

This procedure configures a policy map to set the MPLS EXP field in imposed label entries.

SUMMARY STEPS

1. **class-map** *class-name*
2. **match ip dscp** *dscp-values*
3. **policy-map** *name*
4. **class** *class-name*
5. **police** *bps* [*burst-normal*] [*burst-max*] **conform-action** *action* **exceed-action** *action* [**violate-action***action*]
6. **interface** *type slot* /*port*
7. **service-policy input** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**class-map** *class-name* **Example:** `Router(config)# class-map IP-AF11`	Specifies the class-map to which packets will be mapped (matched). Creates a traffic class.
Step 2	**match ip dscp** *dscp-values* **Example:** `Router(config-c-map)# match ip dscp 4`	Uses the DSCP values as the match criteria for control plane traffic and other traffic that will be transmitted as IP.
Step 3	**policy-map** *name* **Example:** `Router(config)# policy-map set-MPLS-PHB`	Configures the QoS policy for packets that match the class or classes.
Step 4	**class** *class-name* **Example:** `Router(config-p-map)# class IP-AF11`	Associates the traffic class with the service policy.

	Command or Action	Purpose
Step 5	**police** *bps* [*burst-normal*] [*burst-max*] **conform-action** *action* **exceed-action** *action* [**violate-action***action*] **Example:** `Router(config-p-map-c)# police 8000 conform-action set-mpls-experimental-imposition-transmit 4 exceed-action set-mpls-experimental-imposition-transmit 2`	Configures the Traffic Policing feature, including the following: • Action to take on packets that conform to the rate limit specified in the SLA (service level agreement) • Action to take on packets that exceed the rate limit specified in the SLA At the action field, enter **set-mpls-experimental-imposition** *value*, where *value*is the value to which the MPLS EXP field will be set.
Step 6	**interface** *type slot* /*port* **Example:** `Router(config)# interface ethernet 3/0`	Configures an interface type for Cisco series 7200 and Cisco series 7500 routers. Specifies the type of interface to be configured, the port, connector, or interface card number, and the backplane slot number.
Step 7	**service-policy input** *name* **Example:** `Router(config-if)# service-policy input set-MPLS-PHB`	Attaches a QoS policy to an interface and specifies that policies should be applied on packets coming into the interface.

Ingress CE Router--PE Facing Interface

This procedure classifies packets based on their MPLS EXP field and provides appropriate discard and scheduling treatments.

SUMMARY STEPS

1. **class-map match-any** *class-name*
2. **match mpls experimental topmost** *mpls-values*
3. **match ip dscp** *dscp-values*
4. **policy-map** *name*
5. **class** *class-name*
6. **bandwidth** {*bandwidth-kbps* | **percent** *percent*}
7. **random-detect**
8. **interface** *type slot/port*
9. **service-policy output** *name*
10. **mpls ip encapsulate explicit-null**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**class-map match-any** *class-name* **Example:** `Router(config)# class-map match-any MPLS-AF1`	Specifies that packets must meet one of the match criteria to be considered a member of the traffic class.
Step 2	**match mpls experimental topmost** *mpls-values* **Example:** `Router(config-c-map)# match mpls experimental topmost 2 4`	Matches up to eight MPLS EXP field values. Puts matching packets into the same class.
Step 3	**match ip dscp** *dscp-values* **Example:** `Router(config-c-map)# match ip dscp 4`	Uses the DSCP values as the match criteria for control plane traffic and other traffic that will be transmitted as IP.
Step 4	**policy-map** *name* **Example:** `Router(config)# policy-map output-qos`	Configures the QoS policy for packets that match the class or classes.
Step 5	**class** *class-name* **Example:** `Router(config-p-map)# class MPLS-AF1`	Associates the traffic class with the service policy.
Step 6	**bandwidth** {*bandwidth-kbps* | **percent** *percent*} **Example:** `Router(config-p-map-c)# bandwidth percent 40`	Specifies the minimum bandwidth guarantee to a traffic class. You can specify the minimum bandwidth guarantee in kilobits per second or by percent of the overall bandwidth.
Step 7	**random-detect** **Example:** `Router(config-p-map-c)# random-detect`	Applies WRED to the policy based on the IP Precedence or the MPLS EXP field value.
Step 8	**interface** *type slot/port* **Example:** `Router(config)# interface ethernet 3/0`	Configures an interface type for Cisco series 7200 and Cisco series 7500 routers. Specifies the type of interface to be configured, the port, connector, or interface card number, and the backplane slot number.

	Command or Action	Purpose
Step 9	**service-policy output** *name* **Example:** `Router(config-if)# service-policy output output-qos`	Attaches a QoS policy to an interface and specifies that policies should be applied on packets leaving the interface.
Step 10	**mpls ip encapsulate explicit-null** **Example:** `Router(config-if)# mpls ip encapsulate explicit-null`	Encapsulates with an explicit NULL label header all packets forwarded from the interface or subinterface.

Ingress PE Router--P Facing Interface

In this procedure, the default label swap behavior copies the received MPLS EXP field value to the output MPLS EXP field.

SUMMARY STEPS

1. **class-map** *class-name*
2. **match mpls experimental topmost** *mpls-values*
3. **policy-map** *name*
4. **class** *class-name*
5. **bandwidth** {*bandwidth-kbps* | **percent** *percent*}
6. **random-detect**
7. **interface** *type slot/port*
8. **service-policy output** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**class-map** *class-name* **Example:** `Router(config)# class-map MPLS-AF1`	Specifies the class-map to which packets will be mapped (matched). Creates a traffic class.

	Command or Action	Purpose
Step 2	**match mpls experimental topmost** *mpls-values* **Example:** `Router(config-c-map)# match mpls experimental topmost 2 4`	Specifies the MPLS values to use as match criteria against which packets are checked to determine if they belong to the class.
Step 3	**policy-map** *name* **Example:** `Router(config)# policy-map output-qos`	Configures the QoS policy for packets that match the class or classes.
Step 4	**class** *class-name* **Example:** `Router(config-p-map)# class MPLS-AF1`	Associates the traffic class with the service policy.
Step 5	**bandwidth** {*bandwidth-kbps* | **percent** *percent*} **Example:** `Router(config-p-map-c)# bandwidth percent 40`	Specifies the minimum bandwidth guarantee to a traffic class. You can specify the minimum bandwidth guarantee in kilobits per second or by percent of the overall bandwidth.
Step 6	**random-detect** **Example:** `Router(config-p-map-c)# random-detect`	Applies WRED to the policy based on the IP Precedence or the MPLS EXP field value.
Step 7	**interface** *type slot/port* **Example:** `Router(config)# interface ethernet 3/0`	Configures an interface type for Cisco series 7200 and Cisco series 7500 routers. Specifies the type of interface to be configured, the port, connector, or interface card number, and the backplane slot number.
Step 8	**service-policy output** *name* **Example:** `Router(config-if)# service-policy output output-qos`	Attaches a QoS policy to an interface and specifies that policies should be applied on packets leaving the interface.

P Router--P Facing Interface

This procedure classifies packets based on their MPLS EXP field and provides appropriate discard and scheduling treatments.

SUMMARY STEPS

1. **class-map** *class-name*
2. **match mpls experimental topmost** *mpls-values*
3. **policy-map** *name*
4. **class** *class-name*
5. **set mpls experimental topmost** *value*
6. **bandwidth** {*bandwidth-kbps* | **percent** *percent*}
7. **random-detect**
8. **interface** *type slot/port*
9. **service-policy output** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**class-map** *class-name* **Example:** `Router(config)# class-map MPLS-AF1`	Specifies the class-map to which packets will be mapped (matched). Creates a traffic class.
Step 2	**match mpls experimental topmost** *mpls-values* **Example:** `Router(config-c-map)# match mpls experimental topmost 2 4`	Specifies the MPLS EXP field values used as a match criteria against which packets are checked to determine if they belong to the class.
Step 3	**policy-map** *name* **Example:** `Router(config)# policy-map output-qos`	Configures the QoS policy for packets that match the class or classes.
Step 4	**class** *class-name* **Example:** `Router(config-p-map)# class MPLS-AF1`	Associates the traffic class with the service policy.
Step 5	**set mpls experimental topmost** *value* **Example:** `Router(config-p-map-c)# set mpls experimental topmost 3`	Sets the MPLS EXP field value in the topmost MPLS label header at the input and/or output interfaces. This command is optional.

	Command or Action	Purpose
Step 6	**bandwidth** {*bandwidth-kbps* | **percent** *percent*} **Example:** `Router(config-p-map-c)# bandwidth percent 40`	Specifies the minimum bandwidth guarantee to a traffic class. You can specify the minimum bandwidth guarantee in kilobits per second or by percent of the overall bandwidth.
Step 7	**random-detect** **Example:** `Router(config-p-map-c)# random-detect`	Applies WRED to the policy based on the IP Precedence or the MPLS EXP field value.
Step 8	**interface** *type slot/port* **Example:** `Router(config)# interface ethernet 3/0`	Configures an interface type for Cisco series 7200 and Cisco series 7500 routers. Specifies the type of interface to be configured, the port, connector, or interface card number, and the backplane slot number.
Step 9	**service-policy output** *name* **Example:** `Router(config-if)# service-policy output output-qos`	Attaches a QoS policy to an interface and specifies that policies should be applied on packets leaving the interface.

Egress PE Router--P Facing Interface

In this procedure, the qos-group and discard-class convey a packet's PHB to the output interface. The qos-group and discard-class will be used for QoS classification and then will be discarded. The output IP packet's ToS field will not be overwritten.

SUMMARY STEPS

1. **class-map** *class-name*
2. **match mpls experimental topmost** *mpls-values*
3. **policy-map** *name*
4. **class** *class-name*
5. **set qos-group** *qos-group-value*
6. **set discard-class** *value*
7. **interface** *type slot/port*
8. **service-policy input** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**class-map** *class-name* **Example:** `Router(config)# class-map MPLS-AF11`	Specifies the class-map to which packets will be mapped (matched). Creates a traffic class.
Step 2	**match mpls experimental topmost** *mpls-values* **Example:** `Router(config-c-map)# match mpls experimental topmost 4`	Specifies the packet characteristics that will be matched to the class.
Step 3	**policy-map** *name* **Example:** `Router(config)# policy-map set-PHB`	Configures the QoS policy for packets that match the class or classes.
Step 4	**class** *class-name* **Example:** `Router(config-p-map)# class MPLS-AF11`	Associates the traffic class with the service policy.
Step 5	**set qos-group** *qos-group-value* **Example:** `Router(config-p-map-c)# set qos-group 1`	Sets a group ID that can be used later to classify packets. Valid values are from 0 to 99.
Step 6	**set discard-class** *value* **Example:** `Router(config-p-map-c)# set discard-class 1`	Marks a packet with a discard-class value. Specifies the type of traffic that will be dropped when there is congestion. Valid values are from 0 to 7.
Step 7	**interface** *type slot/port* **Example:** `Router(config)# interface ethernet 3/0`	Configures an interface type for Cisco series 7200 and Cisco series 7500 routers. Specifies the type of interface to be configured, the port, connector, or interface card number, and the backplane slot number.
Step 8	**service-policy input** *name* **Example:** `Router(config-if)# service-policy input set-PHB`	Attaches a QoS policy to an interface and specifies that policies should be applied on packets coming into the interface.

Egress PE Router--Customer Facing Interface

This procedure classifies a packet according to the QoS group ID and determines a packet's discard treatment according to the discard-class attribute.

SUMMARY STEPS

1. **class-map** *class-name*
2. **match qos-group** *qos-group-value*
3. **policy-map** *name*
4. **class** *class-name*
5. **bandwidth** {*bandwidth-kbps* | **percent** *percent*}
6. **random-detect discard-class-based**
7. **interface** *type slot/port*
8. mpls ip
9. **service-policy output** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**class-map** *class-name* **Example:** `Router(config)# class-map Local-AF1`	Specifies the class-map to which packets will be mapped (matched). Creates a traffic class.
Step 2	**match qos-group** *qos-group-value* **Example:** `Router(config-c-map)# match qos-group 1`	Identifies a specified QoS group value as a match criteria.
Step 3	**policy-map** *name* **Example:** `Router(config)# policy-map output-qos`	Configures the QoS policy for packets that match the class or classes.
Step 4	**class** *class-name* **Example:** `Router(config-p-map)# class Local-AF1`	Associates the traffic class with the service policy.

	Command or Action	Purpose
Step 5	**bandwidth** {*bandwidth-kbps* | **percent** *percent*} **Example:** `Router(config-p-map-c)# bandwidth percent 40`	Specifies the minimum bandwidth guarantee to a traffic class. You can specify the minimum bandwidth guarantee in kilobits per second or by percent of the overall bandwidth.
Step 6	**random-detect discard-class-based** **Example:** `Router(config-p-map-c)# random-detect discard-class-based`	Bases WRED on the discard class value of a packet.
Step 7	**interface** *type slot/port* **Example:** `Router(config)# interface ethernet 3/0`	Configures an interface type for Cisco series 7200 and Cisco series 7500 routers. Specifies the type of interface to be configured, the port, connector, or interface card number, and the backplane slot number.
Step 8	mpls ip **Example:** `Router(config-if)# mpls ip`	Enables MPLS forwarding of IP version 4 (IPv4) packets along normally routed paths for a particular interface. **Note** You must issue the **mpls ip**command on this interface to receive packets with an explicit-NULL label from the CE router. The **mpls ip** command is not configured on the CE router's interface connected to this interface and therefore no LDP nor other label distribution protocol sessions will be established on this link.
Step 9	**service-policy output** *name* **Example:** `Router(config-if)# service-policy output output-qos`	Attaches a QoS policy to an interface and specifies that policies should be applied on packets leaving the interface.

Configuring Short Pipe Mode

This section describes how to configure the following:

For examples, see the Short Pipe Mode Configuration Example, on page 92.

Note The steps that follow show one way, but not the only way, to configure Short Pipe mode.

Ingress PE Router--Customer Facing Interface

This procedure configures a policy map to set the MPLS EXP field in imposed label entries.

SUMMARY STEPS

1. **class-map** *class-name*
2. **match ip dscp** *dscp-values*
3. **policy-map** *name*
4. **class** *class-name*
5. **police** *bps* [*burst-normal*] [*burst-max*] **conform-action** *action* **exceed-action** *action* [**violate-action** *action*]
6. **interface** *type slot/port*
7. **service-policy input** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**class-map** *class-name* **Example:** `Router(config)# class-map IP-AF11`	Specifies the class-map to which packets will be mapped (matched). Creates a traffic class.
Step 2	**match ip dscp** *dscp-values* **Example:** `Router(config-c-map)# match ip dscp 4`	Uses the DSCP values as the match criteria.
Step 3	**policy-map** *name* **Example:** `Router(config)# policy-map set-MPLS-PHB`	Configures the QoS policy for packets that match the class or classes.
Step 4	**class** *class-name* **Example:** `Router(config-p-map)# class IP-AF11`	Associates the traffic class with the service policy.
Step 5	**police** *bps* [*burst-normal*] [*burst-max*] **conform-action** *action* **exceed-action** *action* [**violate-action** *action*] **Example:** `Router(config-p-map-c)# police 8000 conform-action`	Configures the Traffic Policing feature, including the following: • Action to take on packets that conform to the rate limit specified in the SLA. • Action to take on packets that exceed the rate limit specified in the SLA.

	Command or Action	Purpose
	`set-mpls-experimental-imposition-transmit 4 exceed-action set-mpls-experimental-imposition-transmit 2`	At the action field, enter **set-mpls-experimental-imposition** *value*, where *value*is the value to which the MPLS EXP field will be set.
Step 6	**interface** *type slot/port* **Example:** `Router(config)# interface ethernet 3/0`	Configures an interface type for Cisco series 7200 and Cisco series 7500 routers. Specifies the type of interface to be configured, the port, connector, or interface card number, and the backplane slot number.
Step 7	**service-policy input** *name* **Example:** `Router(config-if)# service-policy input set-MPLS-PHB`	Attaches a QoS policy to an interface and specifies that policies should be applied on packets coming into the interface.

Ingress PE Router--P Facing Interface

This procedure classifies packets based on their MPLS EXP field and provides appropriate discard and scheduling treatments.

SUMMARY STEPS

1. **class-map** *class-name*
2. **match mpls experimental topmost** *mpls-values*
3. **policy-map** *name*
4. **class** *class-name*
5. **bandwidth** {*bandwidth-kbps* | **percent** *percent*}
6. **random-detect**
7. **interface** *type slot/port*
8. **service-policy output** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**class-map** *class-name* **Example:** `Router(config)# class-map MPLS-AF1`	Specifies the class-map to which packets will be mapped (matched). Creates a traffic class.

	Command or Action	Purpose
Step 2	**match mpls experimental topmost** *mpls-values* **Example:** `Router(config-c-map)# match mpls experimental topmost 2 4`	Specifies the MPLS EXP field values used as a match criteria against which packets are checked to determine if they belong to the class.
Step 3	**policy-map** *name* **Example:** `Router(config)# policy-map output-qos`	Configures the QoS policy for packets that match the class or classes.
Step 4	**class** *class-name* **Example:** `Router(config-p-map)# class MPLS-AF1`	Associates the traffic class with the service policy.
Step 5	**bandwidth** {*bandwidth-kbps* | **percent** *percent*} **Example:** `Router(config-p-map-c)# bandwidth percent 40`	Specifies the minimum bandwidth guarantee to a traffic class. You can specify the minimum bandwidth guarantee in kilobits per second or by percent of the overall bandwidth.
Step 6	**random-detect** **Example:** `Router(config-p-map-c)# random-detect`	Enables a WRED drop policy for a traffic class that has a bandwidth guarantee.
Step 7	**interface** *type slot/port* **Example:** `Router(config)# interface ethernet 3/0`	Configures an interface type for Cisco series 7200 and Cisco series 7500 routers. Specifies the type of interface to be configured, the port, connector, or interface card number, and the backplane slot number.
Step 8	**service-policy output** *name* **Example:** `Router(config-if)# service-policy output-qos`	Attaches a QoS policy to an interface and specifies that policies should be applied on packets leaving the interface.

P Router--P Facing Interface

This procedure classifies packets based on their MPLS EXP field and provides appropriate discard and scheduling treatments.

SUMMARY STEPS

1. **class-map** *class-name*
2. **match mpls experimental topmost** *mpls-values*
3. **policy-map** *name*
4. **class** *class-name*
5. **bandwidth** {*bandwidth-kbps* | **percent** *percent*}
6. **random-detect**
7. **interface** *type slot/port*
8. **service-policy output** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**class-map** *class-name* **Example:** `Router(config)# class-map MPLS-AF1`	Specifies the class-map to which packets will be mapped (matched). Creates a traffic class.
Step 2	**match mpls experimental topmost** *mpls-values* **Example:** `Router(config-c-map)# match mpls experimental topmost 2 4`	Specifies the MPLS EXP field values used as a match criteria against which packets are checked to determine if they belong to the class.
Step 3	**policy-map** *name* **Example:** `Router(config)# policy-map output-qos`	Configures the QoS policy for packets that match the class or classes.
Step 4	**class** *class-name* **Example:** `Router(config-p-map)# class MPLS-AF1`	Associates the traffic class with the service policy.
Step 5	**bandwidth** {*bandwidth-kbps* | **percent** *percent*} **Example:** `Router(config-p-map-c)# bandwidth percent 40`	Specifies the minimum bandwidth guarantee to a traffic class. You can specify the minimum bandwidth guarantee in kilobits per second or by percent of the overall bandwidth.
Step 6	**random-detect** **Example:** `Router(config-p-map-c)# random-detect`	Applies WRED to the policy based on the IP Precedence or the MPLS EXP field value.

	Command or Action	Purpose
Step 7	**interface** *type slot/port* **Example:** `Router(config)# interface ethernet 3/0`	Configures an interface type for Cisco series 7200 and Cisco series 7500 routers. Specifies the type of interface to be configured, the port, connector, or interface card number, and the backplane slot number.
Step 8	**service-policy output** *name* **Example:** `Router(config-if)# service-policy output output-qos`	Attaches a QoS policy to an interface and specifies that policies should be applied on packets leaving the interface.

Egress PE Router--Customer Facing Interface

This procedure classifies a packet based on its IP DSCP value and provides appropriate discard and scheduling treatments.

SUMMARY STEPS

1. **class-map** *class-name*
2. **match ip dscp** *dscp-values*
3. **policy-map** *name*
4. **class** *class-name*
5. **bandwidth** {*bandwidth-kbps* | **percent** *percent*}
6. **random-detect dscp-based**
7. **interface** *type slot/port*
8. **service-policy output** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**class-map** *class-name* **Example:** `Router(config)# class-map IP-AF1`	Specifies the class-map to which packets will be mapped (matched). Creates a traffic class.
Step 2	**match ip dscp** *dscp-values* **Example:** `Router(config-c-map)# match ip dscp 4 0`	Uses the DSCP values as the match criteria.

	Command or Action	Purpose
Step 3	**policy-map** *name* **Example:** `Router(config)# policy-map output-qos`	Configures the QoS policy for packets that match the class or classes.
Step 4	**class** *class-name* **Example:** `Router(config-p-map)# class AF1`	Associates the traffic class with the service policy.
Step 5	**bandwidth** {*bandwidth-kbps* | **percent** *percent*} **Example:** `Router(config-p-map-c)# bandwidth percent 40`	Specifies the minimum bandwidth guarantee to a traffic class. You can specify the minimum bandwidth guarantee in kilobits per second or by percent of the overall bandwidth.
Step 6	**random-detect dscp-based** **Example:** `Router(config-p-map-c)# random-detect dscp-based`	Enables a WRED drop policy for a traffic class that has a bandwidth guarantee.
Step 7	**interface** *type slot/port* **Example:** `Router(config)# interface ethernet 3/0`	Configures an interface type for Cisco series 7200 and Cisco series 7500 routers. Specifies the type of interface to be configured, the port, connector, or interface card number, and the backplane slot number.
Step 8	**service-policy output** *name* **Example:** `Router(config-if)# service-policy output output-qos`	Attaches a QoS policy to an interface and specifies that policies should be applied on packets leaving the interface.

Configuring Uniform Mode

This section describes how to configure the following:

For examples, see the Uniform Mode Configuration Example, on page 93.

Note The steps that follow show one way, but not the only way, to configure Uniform mode.

Ingress PE Router--Customer Facing Interface

This procedure configures a policy map to set the MPLS EXP field in imposed label entries.

SUMMARY STEPS

1. **class-map** *class-name*
2. **match ip dscp** *dscp-values*
3. **policy-map** *name*
4. **class** *class-name*
5. **police** *bps* [*burst-normal*] [*burst-max*] **conform-action** *action* **exceed-action** *action* [**violate-action** *action*]
6. **interface** *type slot/port*
7. **service-policy input** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**class-map** *class-name* **Example:** `Router(config)# class-map IP-AF11`	Specifies the class-map to which packets will be mapped (matched). Creates a traffic class.
Step 2	**match ip dscp** *dscp-values* **Example:** `Router(config-c-map)# match ip dscp 4`	Uses the DSCP values as the match criteria.
Step 3	**policy-map** *name* **Example:** `Router(config)# policy-map set-MPLS-PHB`	Configures the QoS policy for packets that match the class or classes.
Step 4	**class** *class-name* **Example:** `Router(config-p-map)# class IP-AF11`	Associates the traffic class with the service policy.
Step 5	**police** *bps* [*burst-normal*] [*burst-max*] **conform-action** *action* **exceed-action** *action* [**violate-action** *action*] **Example:** `Router(config-p-map-c)# police 8000 conform-action`	Configures the Traffic Policing feature, including the following: • Action to take on packets that conform to the rate limit specified in the SLA. • Action to take on packets that exceed the rate limit specified in the SLA.

	Command or Action	Purpose
	`set-mpls-experimental-imposition-transmit 3 exceed-action set-mpls-experimental-imposition-transmit 2`	At the action field, enter **set-mpls-experimental-imposition** *value,*where *value* is the value to which the MPLS EXP field will be set.
Step 6	**interface** *type slot/port* **Example:** `Router(config)# interface ethernet 3/0`	Configures an interface type for Cisco series 7200 and Cisco series 7500 routers. Specifies the type of interface to be configured, the port, connector, or interface card number, and the backplane slot number.
Step 7	**service-policy input** *name* **Example:** `Router(config-if)# service-policy input set-MPLS-PHB`	Attaches a QoS policy to an interface and specifies that policies should be applied on packets coming into the interface.

Ingress PE Router--P Facing Interface

This procedure classifies packets based on their MPLS EXP field and provides appropriate discard and scheduling treatments.

SUMMARY STEPS

1. **class-map** *class-name*
2. **match mpls experimental topmost** *mpls-values*
3. **policy-map** *name*
4. **class** *class-name*
5. **bandwidth** {*bandwidth-kbps* | **percent** *percent*}
6. **random-detect**
7. **interface** *type slot/port*
8. **service-policy output** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**class-map** *class-name* **Example:** `Router(config)# class-map MPLS-AF1`	Specifies the class-map to which packets will be mapped (matched). Creates a traffic class.

	Command or Action	Purpose
Step 2	**match mpls experimental topmost** *mpls-values* **Example:** `Router(config-c-map)# match mpls experimental topmost 2 3`	Specifies the MPLS EXP field values used as a match criteria against which packets are checked to determine if they belong to the class.
Step 3	**policy-map** *name* **Example:** `Router(config)# policy-map output-qos`	Configures the QoS policy for packets that match the class or classes.
Step 4	**class** *class-name* **Example:** `Router(config-p-map)# class MPLS-AF1`	Associates the traffic class with the service policy.
Step 5	**bandwidth** {*bandwidth-kbps* | **percent** *percent*} **Example:** `Router(config-p-map-c)# bandwidth percent 40`	Specifies the minimum bandwidth guarantee to a traffic class. You can specify the minimum bandwidth guarantee in kilobits per second or by percent of the overall bandwidth.
Step 6	**random-detect** **Example:** `Router(config-p-map-c)# random-detect`	Enables a WRED drop policy for a traffic class that has a bandwidth guarantee.
Step 7	**interface** *type slot/port* **Example:** `Router(config)# interface ethernet 3/0`	Configures an interface type for Cisco series 7200 and Cisco series 7500 routers. Specifies the type of interface to be configured, the port, connector, or interface card number, and the backplane slot number.
Step 8	**service-policy output** *name* **Example:** `Router(config-if)# service-policy output-qos`	Attaches a QoS policy to an interface and specifies that policies should be applied on packets leaving the interface.

P Router--Upstream P Facing Interface

This procedure classifies a packet based on the MPLS EXP field and sets the QoS group ID.

SUMMARY STEPS

1. **class-map** *class-name*
2. **match mpls experimental topmost** *mpls-values*
3. **policy-map** *name*
4. **class** *class-name*
5. **set qos-group mpls experimental topmost**
6. **interface** *type slot/port*
7. **service-policy input** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**class-map** *class-name* **Example:** `Router(config)# class-map mpls-in`	Specifies the class-map to which packets will be mapped (matched). Creates a traffic class.
Step 2	**match mpls experimental topmost** *mpls-values* **Example:** `Router(config-c-map)# match mpls experimental topmost 4 5`	Specifies the MPLS EXP field values used as a match criteria against which packets are checked to determine if they belong to the class.
Step 3	**policy-map** *name* **Example:** `Router(config)# policy-map policy2`	Configures the QoS policy for packets that match the class or classes.
Step 4	**class** *class-name* **Example:** `Router(config-p-map)# class mpls-in`	Associates the traffic class with the service policy.
Step 5	**set qos-group mpls experimental topmost** **Example:** `Router(config-p-map-c)# set qos-group mpls experimental topmost`	Copies the MPLS EXP topmost field value into the QoS group ID. For more information, refer to *Enhanced Packet Marking* , Release 12.2(13)T.
Step 6	**interface** *type slot/port* **Example:** `Router(config)# interface ethernet 3/0`	Configures an interface type for Cisco series 7200 and Cisco series 7500 routers. Specifies the type of interface to be configured, the port, connector, or interface card number, and the backplane slot number.

	Command or Action	Purpose
Step 7	**service-policy input** *name* **Example:** `Router(config-if)# service-policy input policy2`	Attaches a QoS policy to an interface and specifies that policies should be applied on packets coming into the interface.

P Router--Downstream P Facing Interface

This procedure matches packets based on their QoS ID and sets the MPLS EXP field in the topmost label header to the QoS group ID.

SUMMARY STEPS

1. **class-map** *class-name*
2. **match qos-group** *qos-group-value*
3. **policy-map** *name*
4. **class** *class-name*
5. **set mpls experimental topmost qos-group**
6. **bandwidth** {*bandwidth-kbps* | **percent** *percent*}
7. **random-detect**
8. **interface** *type slot/port*
9. **service-policy output** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**class-map** *class-name* **Example:** `Router(config)# class-map qos-group-out`	Specifies the class-map to which packets will be mapped (matched). Creates a traffic class.
Step 2	**match qos-group** *qos-group-value* **Example:** `Router(config-c-map)# match qos-group 4`	Identifies a specified QoS group value as a match criterion.

	Command or Action	Purpose
Step 3	**policy-map** *name* **Example:** `Router(config)# policy-map policy3`	Configures the QoS policy for packets that match the class or classes.
Step 4	**class** *class-name* **Example:** `Router(config-p-map)# class qos-group-out`	Associates the traffic class with the service policy.
Step 5	**set mpls experimental topmost qos-group** **Example:** `Router(config-p-map-c)# set mpls experimental topmost qos-group`	Copies the QoS group ID into the MPLS EXP field of the topmost label header.
Step 6	**bandwidth** {*bandwidth-kbps* | **percent** *percent*} **Example:** `Router(config-p-map-c)# bandwidth percent 40`	Specifies the minimum bandwidth guarantee to a traffic class. You can specify the minimum bandwidth guarantee in kilobits per second or by percent of the overall bandwidth.
Step 7	**random-detect** **Example:** `Router(config-p-map-c)# random-detect`	Applies WRED to the policy based on the IP Precedence or the MPLS EXP field value.
Step 8	**interface** *type slot/port* **Example:** `Router(config)# interface ethernet 3/1`	Configures an interface type for Cisco series 7200 and Cisco series 7500 routers. Specifies the type of interface to be configured, the port, connector, or interface card numbers, and the backplane slot number.
Step 9	**service-policy output** *name* **Example:** `Router(config-if)# service-policy output policy3`	Attaches a QoS policy to an interface and specifies that policies should be applied on packets coming into the interface.

Egress PE Router--P Facing Interface

This procedure classifies a packet based on the MPLS EXP field and sets the QoS group ID.

SUMMARY STEPS

1. **class-map** *class-name*
2. **match mpls experimental topmost** *mpls-values*
3. **policy-map** *name*
4. **class** *class-name*
5. **set qos-group mpls experimental topmost**
6. **interface** *type slot /port*
7. **service-policy input** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**class-map** *class-name* **Example:** `Router(config)# class-map mpls-in`	Specifies the class-map to which packets will be mapped (matched). Creates a traffic class.
Step 2	**match mpls experimental topmost** *mpls-values* **Example:** `Router(config-c-map)# match mpls experimental topmost 4 5`	Specifies the MPLS EXP field values used as a match criteria against which packets are checked to determine if they belong to the class.
Step 3	**policy-map** *name* **Example:** `Router(config)# policy-map foo`	Configures the QoS policy for packets that match the class or classes.
Step 4	**class** *class-name* **Example:** `Router(config-p-map)# class mpls-in`	Associates the traffic class with the service policy.
Step 5	**set qos-group mpls experimental topmost** **Example:** `Router(config-p-map)# set qos-group mpls experimental topmost`	Copies the MPLS EXP topmost field value into the QoS group ID.
Step 6	**interface** *type slot /port* **Example:** `Router(config)# interface ethernet 3/0`	Configures an interface type for Cisco series 7200 and Cisco series 7500 routers. Specifies the type of interface to be configured, the port, connector, or interface card numbers, and the backplane slot number.

	Command or Action	Purpose
Step 7	**service-policy input** *name* **Example:** `Router(config-if)# service-policy input foo`	Attaches a QoS policy to an interface and specifies that policies should be applied on packets coming into the interface.

Egress PE Router--Customer Facing Interface

This procedure matches packets based on their QoS ID and sets the IP Precedence field to the QoS group ID.

SUMMARY STEPS

1. **class-map** *class-name*
2. **match qos-group** *qos-group-value*
3. **policy-map** *name*
4. **class** *class-name*
5. **set precedence qos-group**
6. **bandwidth** {*bandwidth-kbps* | **percent** *percent*}
7. **random-detect**
8. **interface** *type slot /port*
9. **service-policy output** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**class-map** *class-name* **Example:** `Router(config)# class-map qos-out`	Specifies the class-map to which packets will be mapped (matched). Creates a traffic class.
Step 2	**match qos-group** *qos-group-value* **Example:** `Router(config-c-map)# match qos-group 4`	Identifies a specified QoS group value as a match criterion.
Step 3	**policy-map** *name* **Example:** `Router(config)# policy-map foo-out`	Configures the QoS policy for packets that match the class or classes.

	Command or Action	Purpose
Step 4	**class** *class-name* **Example:** `Router(config-p-map)# class qos-out`	Associates the traffic class with the service policy.
Step 5	**set precedence qos-group** **Example:** `Router(config-p-map-c)# set precedence qos-group`	Sets the Precedence value in the packet header.
Step 6	**bandwidth** {*bandwidth-kbps* | **percent** *percent*} **Example:** `Router(config-p-map-c)# bandwidth percent 40`	Specifies the minimum bandwidth guarantee to a traffic class. You can specify the minimum bandwidth guarantee in kilobits per second or by percent of the overall bandwidth.
Step 7	**random-detect** **Example:** `Router(config-p-map-c)# random-detect`	Applies WRED to the policy based on the IP Precedence or the MPLS EXP field value.
Step 8	**interface** *type slot /port* **Example:** `Router(config)# interface ethernet 3/1`	Configures an interface type for Cisco series 7200 and Cisco series 7500 routers. Specifies the type of interface to be configured, the port, connector, or interface card numbers, and the backplane slot number.
Step 9	**service-policy output** *name* **Example:** `Router(config-if)# service-policy output foo-out`	Attaches a QoS policy to an interface and specifies that policies should be applied on packets coming into the interface.

Verifying MPLS DiffServ Tunneling Mode Support

- On PE routers, the **show policy-map interface** command displays the discard-class-based WRED in the output.
- In Short Pipe mode, the **show policy-map interface** command looks for the **set mpls experimental imposition** command.

Troubleshooting Tips

- The IP QoS marking should not change in the service provider's network.
- QoS statistics should indicate that packets were scheduled in the correct classes.

Configuration Examples for MPLS DiffServ Tunneling Modes

Note You can configure only one tunneling mode.

- The examples that follow show one way, but not the only way, to configure the tunneling modes.

Pipe Mode with an Explicit NULL LSP Configuration Example

Ingress CE Router--Customer Facing Interface

In this example, packets are matched to class-map IP-AF11. The DSCP value 4 is used as the match criterion to determine whether a packet belongs to that class. Packets that are conforming have their MPLS EXP field set to 4. Packets that are out-of-rate have their MPLS EXP field set to 2.

```
class-map IP-AF11
 match ip dscp 4
policy-map set-MPLS-PHB
 class IP-AF11
  police 8000 conform-action set-mpls-experimental-imposition-transmit 4 exceed-action
  set-mpls-experimental-imposition-transmit 2
interface ethernet 3/0
 service-policy input set-MPLS-PHB
```

Ingress CE Router--PE Facing Interface

In this example, MPLS EXP 2 and 4 are matched to class-map MPLS-AF1. Packets that match that class have WRED and WFQ enabled.

```
class-map match-any MPLS-AF1
 match mpls experimental topmost 2 4
 match ip dscp 4
policy-map output-qos
class MPLS-AF1
  bandwidth percent 40
  random-detect
interface ethernet 3/0
 service-policy output output-qos
 mpls ip encapsulate explicit-null
```

Ingress PE Router--P Facing Interface

In this example, the default label swap behavior copies the received MPLS EXP field value to the output MPLS EXP field. Packets that have an MPLS EXP value of 2 and 4 are matched to class-map MPLS-AF1. Packets that match that class have WRED and WFQ enabled.

```
class-map MPLS-AF1
 match mpls experimental topmost 2 4
policy-map output-qos
 class MPLS-AF1
   bandwidth percent 40
   random-detect
interface ethernet 3/0
 service-policy output output-qos
```

P Router--P Facing Interface

In this example, packets that have an MPLS EXP value of 2 or 4 are matched to class-map MPLS-AF1. Packets that match that class have WRED and WFQ enabled.

```
class-map MPLS-AF1
 match mpls experimental topmost 2 4
policy-map output-qos
 class MPLS-AF1
  bandwidth percent 40
  random-detect
interface ethernet 3/0
 service-policy output output-qos
```

Egress PE Router--P Facing Interface

In this example, qos-group 1 and discard-class 1 must be set to indicate the packet's PHB. The qos-group and discard-class are used for QoS classification at the output interface.

```
class-map MPLS-AF11
 match mpls experimental topmost 4
class-map MPLS-AF12
 match mpls experimental topmost 2
policy-map set-PHB
 class MPLS-AF11
  set qos-group 1
  set discard-class 1
class MPLS-AF12
  set qos-group 1
  set discard-class 2
interface ethernet 3/0
 service-policy input set-PHB
```

Egress PE Router--Customer Facing Interface

In this example, packets that have a qos-group value of 1 are matched to class-map Local-AF1. Packets that match that class have WRED based on their discard class value applied.

Note You must issue the **mpls ip** command on this interface to receive packets with an explicit-NULL label from the CE router. The **mpls ip** command is not configured on the CE router's interface connected to this interface and therefore no LDP nor other label distribution protocol sessions will be established on this link.

```
class-map Local-AF1
 match qos-group 1
```

```
policy-map output-qos
 class Local-AF1
  bandwidth percent 40
  random-detect discard-class-based
interface ethernet 3/0
 mpls ip
 service-policy output output-qos
```

Short Pipe Mode Configuration Example

Note Short Pipe mode is not configured on CE routers.

Ingress PE Router--Customer Facing Interface

In this example, IP packets are matched to class-map IP-AF11. Packets that are conforming have their MPLS EXP field set to 4. Packets that are out-of-rate have their MPLS EXP field set to 2.

```
class-map IP-AF11
 match ip dscp 4
policy-map set-MPLS-PHB
 class IP-AF11
  police 8000 conform-action set-mpls-experimental-imposition-transmit 4 exceed-action
  set-mpls-experimental-imposition-transmit 2
interface ethernet 3/0
 service-policy input set-MPLS-PHB
```

Ingress PE Router--P Facing Interface

In this example, MPLS EXP 2 and 4 are matched to class-map MPLS-AF1. Packets that match that class have WRED and WFQ enabled.

```
class-map MPLS-AF1
 match mpls experimental topmost 2 4
policy-map output-qos
 class MPLS-AF1
  bandwidth percent 40
  random-detect
interface ethernet 3/0
 service-policy output output-qos
```

P Router--P Facing Interface

In this example, MPLS EXP 2 and 4 are matched to class-map MPLS-AF1. Packets that match that class have WRED and WFQ enabled.

```
class-map MPLS-AF1
 match mpls experimental topmost 2 4
policy-map output-qos
 class MPLS-AF1
  bandwidth percent 40
  random-detect
interface ethernet 3/0
 service-policy output output-qos
```

Egress PE Router--Customer Facing Interface

In this example, the egress PE router transmits IP packets. Packets are matched to class-map IP-AF1. Packets that match that class have WRED and WFQ enabled.

```
class-map IP-AF1
 match ip dscp 4 0
policy-map output-qos
 class AF1
  bandwidth percent 40
  random-detect dscp-based
interface ethernet 3/0
 service-policy output output-qos
```

Uniform Mode Configuration Example

Ingress PE Router--Customer Facing Interface

In this example, IP packets are matched to class-map IP-AF11. Packets that are conforming have their MPLS EXP field set to 3. Packets that are out-of-rate have their MPLS EXP field set to 2.

```
class-map IP-AF11
 match ip dscp 4
policy-map set-MPLS-PHB
 class IP-AF11
  police 8000 conform-action set-mpls-experimental-imposition-transmit 3 exceed-action
  set-mpls-experimental-imposition-transmit 2
interface ethernet 3/0
 service-policy input set-MPLS-PHB
```

Ingress PE Router--P Facing Interface

In this example, MPLS EXP 2 and 3 are matched to class-map MPLS-AF1. Packets that match that class have WRED and WFQ enabled.

```
class-map MPLS-AF1
 match mpls experimental topmost 2 3
policy-map output-qos
 class MPLS-AF1
  bandwidth percent 40
  random-detect
interface ethernet 3/0
 service-policy output output-qos
```

P Router--Upstream P Facing Interface

At the penultimate P router's input interface where the IGP label is popped, the EXP field value in the IGP label is copied to the QoS group ID. Suppose the MPLS EXP field value in the IGP label was recolored in the core to 4 or 5. In this example, MPLS EXP values 4 and 5 are matched to class-map mpls-in. For packets that match that class, the MPLS EXP value in the IGP label is copied to the QoS group ID.

```
class-map mpls-in
 match mpls experimental topmost 4 5
policy-map policy2
 class mpls-in
  set qos-group mpls experimental topmost
interface ethernet 3/0
 service-policy input policy2
```

P Router--Downstream P Facing Interface

In this example, QoS group IDs 4 and 5 are matched to class-map qos-group-out. For packets that match that class, the MPLS EXP field in the topmost outgoing label is set to the QoS group ID.

```
class-map qos-group-out
 match qos-group 4
 match qos-group 5
policy-map policy3
 class qos-group-out
  set mpls experimental topmost qos-group
  bandwidth percent 40
  random-detect
interface ethernet 3/1
 service-policy output policy3
```

Egress PE Router--P Facing Interface

In this example, packets with MPLS EXP values 4 or 5 are matched to class-map mpls-in. The EXP field value from the label header is copied to the QoS group ID.

```
class-map mpls-in
 match mpls experimental topmost 4 5
policy-map foo
 class mpls-in
  set qos-group mpls experimental topmost
interface ethernet 3/0
 service-policy input foo
```

Egress PE Router--Customer Facing Interface

In this example, the egress PE router transmits IP packets. QoS group IDs 4 and 5 are matched into class-map qos-out and the IP Precedence field of those packets is set to the QoS group ID.

```
class-map qos-out
 match qos-group 4
 match qos-group 5
policy-map foo-out
 class qos-out
  set precedence qos-group
  bandwidth percent 40
  random-detect
interface ethernet 3/1
 service-policy output foo-out
```

Additional References

Related Documents

Related Topic	Document Title
MPLS Traffic Engineering	MPL S Configuration Guide
QoS	Quality of Service Configuration Guide

Standards

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

MIBs

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	To locate and download MIBs for selected platforms, Cisco IOS 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 this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for MPLS DiffServ Tunneling Modes

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 4: Feature Information for MPLS DiffServ Tunneling Modes

Feature Name	Releases	Feature Information
MPLS DiffServ Tunneling Modes	12.2(13)T 12.2(18)S 12.2(27)SBA 12.2(27)SBB 12.2(28)SB 12.3(2)T Cisco IOS Release IOS XE 2.1 15.4(1)S	The MPLS DiffServ Tunneling Modes feature allows service providers to manage the QoS that a router will provide to an MPLS packet in an MPLS network. In 12.2(13)T, this feature was introduced. In 12.2(18)S, this feature was integrated. In 12.2(27)SBA, this feature was integrated. In 12.2(27)SBB, this feature was integrated. In 12.2(28)SB, this feature was integrated. In 12.3(2)T, this feature was integrated. In Cisco IOS Release IOS XE 2.1, this feature was integrated. In Cisco IOS Release 15.4(1) S, support was added for the Cisco 901S platform.

Glossary

CE router --customer edge router. A router that is part of a customer network and that interfaces to a provider edge (PE) router.

class --Classifies traffic, such as voice. You define a traffic class with the **class-map** command.

class-map --Defines what you want to match in a packet. For example, a class-map may specify voice packets.

core --The MPLS network. At the edges, there are edge routers.

customer network --A network that is under the control of an end customer. A customer network can use private addresses as defined in RFC 1918. Customer networks are logically isolated from each other and from the service provider's network.

DiffServ --Application-level QoS and traffic management in an architecture that incorporates mechanisms to control bandwidth, delay, jitter, and packet loss. Application traffic can be categorized into multiple classes (aggregates), with QoS parameters defined for each class. A typical arrangement would be to categorize traffic into premium, gold, silver, bronze, and best-effort classes.

DSCP --differentiated services code point, or DiffServ code point. A marker in the header of each IP packet that prompts network routers to apply differentiated grades of service to various packet streams. The value in the IP header indicates which PHB is to be applied to the packet.

discard-class --Local variable used to indicate the discard profile.

E-LSP --An LSP in which the QoS of a packet is determined solely by the MPLS EXP field in the MPLS header. E-LSPs are not supported by ATM-LSRs.

edge router --A router that is at the edge of the network. It defines the boundary of the MPLS network. It receives and transmits packets. Also referred to as edge label switch router and label edge router.

egress router --Router at the edge of the network where packets are leaving.

encapsulation --The wrapping of data in a particular protocol header. For example, Ethernet data is wrapped in a specific Ethernet header before network transit.

explicit null label --A label that just has an EXP value. A value of zero (0) represents the explicit NULL label. This label can only be at the bottom of the label stack. It indicates that the label stack must be popped, and the forwarding of the packet must then be based on the IPv4 header. Sometimes there may be requirements to have a label in the stack when no label is required. If you want to retain the MPLS EXP field to the next hop, you use an explicit null.

ingress router --Router at the edge of the network where packets are being received by the network.

IP Precedence field --The first three bits in the header of IP packets. These bits allow you to specify the QoS for an IP packet.

L-LSP --An LSP where a particular mechanism of implementing QoS using DiffServ is used. An LSP in which routers infer the QoS treatment for MPLS packets from the packet label and the EXP bits (or the CLP bit for cell-mode MPLS). The label is used to encode the class to which a packet belongs and the MPLS EXP field (or the CLP bit for cell-mode MPLS) is used to encode the drop precedence of the packet.

LSR --A router that is part of the MPLS network. An LSR forwards a packet based on the value of a label encapsulated in the packet.

label --A short, fixed-length label that tells switching nodes how to forward data (packets). MPLS associates a label with each route. A label associates a network address with the output interface onto which the packet should be transmitted. In the MPLS network, the next-hop IGP (Interior Gateway Protocol) router always advertises to the preceding IGP router (the upstream router) what label should be placed on the packets. The next-hop BGP (Border Gateway Protocol) router always advertises to the preceding BGP router what label should be placed on the packets.

label disposition --The act of removing the last MPLS label from a packet.

label entry --A label entry contains a label value (which includes labels and other information for forwarding the packet) and an MPLS EXP field (which pertains to the QoS of the packet). When there are two label entries, the top label entry is the IGP (Interior Gateway Protocol) label. The bottom label entry is the BGP (Border Gateway Protocol) label.

label imposition --The act of putting MPLS labels onto a packet for transmission on a label switched path (LSP).

layer management --Ability to apply an additional layer of PHB marking to a packet.

MPLS --Multiprotocol Label Switching. Emerging industry standard upon which label switching is based.

MPLS EXP field --In an MPLS entry, the per-hop behavior (PHB) is marked in the MPLS EXP field within the MPLS label entry.

P router --provider core router.

PE router --provider edge router. A router, at the edge of a service provider's network, that interfaces to CE routers.

penultimate hop popping --Removing a label at the penultimate router. A label is removed and copied to the label that is one lower.

penultimate router --The second-to-last router; that is, the router that is immediately before the egress router.

PHB --per-hop behavior. A unique discard and scheduling behavior that is applied to a packet. The DiffServ treatment (scheduling/dropping) applied by a router to all the packets that are to experience the same DiffServ service.

policing --Limiting the input or output transmission rate of a class of traffic based on user-defined criteria. Policing marks packets by setting the IP precedence value, the qos-group, or the DSCP value.

policy map --Action that is taken if a packet matches what was specified in the class-map. For example, if voice packets were identified and the class-map and voice packets are received, the specified policy map action is taken.

pop --The act of removing a label entry from a packet.

provider network --A backbone network that is under the control of a service provider, and provides transport between customer sites.

push --To put a label entry onto a packet.

QoS --quality of service. Measure of performance for a transmission system that reflects its transmission quality and service availability.

QoS transparency --Method of forwarding packets through a network where the customer's IP marking in the IP packet is preserved.

qos-group --Local variable that indicates the PHB scheduling class (PSC).

rate limiting --See *policing* .

recolor --To change the PHB marking on a packet.

swap --To replace a label entry on a packet.

ToS --type of service. Byte in the IPv4 header.

traffic policy --A traffic policy consists of a traffic class and one or more QoS features. You create a traffic policy by associating the traffic class with one or more QoS features (using the **policy-map** command).

transparency --Preservation of the customer's IP marking in the IP packet.

tunneling --The ability of QoS to be transparent from one edge of a network to the other edge of the network.

VPN --Virtual Private Network. A network that enables IP traffic to use tunneling to travel securely over a public TCP/IP network.

WRED --weighted random early detection. A queuing method that ensures that high-precedence traffic has lower loss rates than other traffic during times of congestion.

INDEX

U

W