MPLS Traffic Engineering - DiffServ Aware (DS-TE)

This guide presents extensions made to Multiprotocol Label Switching Traffic Engineering (MPLS TE) that make it DiffServ aware. Specifically, the bandwidth reservable on each link for constraint-based routing (CBR) purposes can now be managed through at least two bandwidth pools: a *global pool* (also called BC0) and a *sub-pool* (also called BC1). The sub-pool can be limited to a smaller portion of the link bandwidth. Tunnels using the sub-pool bandwidth can then be used in conjunction with MPLS Quality of Service (QoS) mechanisms to deliver guaranteed bandwidth services end-to-end across the network.

Beginning with Cisco IOS Release 12.2(33)SRB, 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.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Feature History

Release	Modification
12.0(11) ST	DS-TE feature introduced.
12.0(14) ST	Support was added for Cisco Series 7500(VIP) platform. Support was added for IS-IS Interior Gateway Protocol.
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).
12.0(22)S	Feature was implemented in Cisco IOS Release 12.0(22)S.
12.2(14)S	Feature was integrated into Cisco IOS Release 12.2(14)S.
12.2(18)S	Feature was implemented in Cisco IOS Release 12.2(18)S.
12.2(18)SXD	Feature was implemented in Cisco IOS Release 12.2(18)SXD.
12.2(28)SB	Feature was implemented in Cisco IOS Release 12.2(28)SB.
12.2(33)SRB	Feature was augmented to include the new IETF-Standard functionality of DS-TE, as described in RFCs 3270, 4124, 4125, and 4127.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

The guide contains the following sections:

Feature Overview

MPLS traffic engineering 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 Traffic Engineering 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 performance in terms of delay, jitter, or loss for the guaranteed traffic.

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**
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.

Table 2 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 Signalling**	**IGP Advertisement**	**RSVP-TE Signalling**
Traditional	No	traditional	traditional	traditional[1]	traditional
Migration	Yes	traditional	traditional	traditional & IETF	traditional & IETF
Liberal IETF	Yes	IETF	traditional & IETF	traditional & IETF	traditional & IETF

[1] Note 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 on page 3).

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.

Benefits

DiffServ-aware Traffic Engineering enables service providers to perform separate admission control and separate route computation for discrete subsets of traffic (for example, voice and data traffic).

Therefore, by combining DS-TE with other IOS 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.

Related Features and Technologies

The DS-TE feature is related to OSPF, IS-IS, RSVP (Resource reSerVation Protocol), QoS, and MPLS traffic engineering. Cisco documentation for all of these features is listed in the next section.

Related Documents

The following sections provide references related to the MPLS Traffic Engineering—DiffServ Aware (DS-TE) feature:

Related Topic	Document Title
IS-IS	• *Cisco IOS IP Routing Protocols Command Reference* • *Configuring a Basic IS-IS Network*
MPLS Traffic Engineering	*Cisco IOS Multiprotocol Label Switching Command Reference*
OSPF	• *Cisco IOS IP Routing Protocols Command Reference* • *OSPF Configuration Task List*
QoS	• *Cisco IOS Quality of Service Solutions Command Reference* • *Quality of Service Overview*
RSVP	• *Cisco IOS Quality of Service Solutions Command Reference* • *Configuring RSVP*

Supported Standards

The traditional (pre-IETF Standard) version of DiffServ-aware MPLS Traffic Engineering conforms to the descriptions given in the following two documents:

- *Requirements for Support of Diff-Serv-aware MPLS Traffic Engineering* by F. Le Faucheur, T. Nadeau, A. Chiu, W. Townsend, D. Skalecki & M. Tatham
- *Protocol Extensions for Support of Diff-Serv-aware MPLS Traffic Engineering* by F. Le Faucheur, T. Nadeau, J. Boyle, K. Kompella, W. Townsend & D. Skalecki.

The IETF Standard for DiffServ-aware MPLS Traffic Engineering is described in the following four documents:

- Multi-Protocol Label Switching (MPLS) Support of Differentiated Services by F. Le Faucheur, L. Wu, B. Davie, P. Vaananen, R. Krishnan, P. Cheval, & J. Heinanen (RFC 3270)
- *Protocol Extensions for Support of Diffserv-aware MPLS Traffic Engineering* ed. by F. Le Faucheur (RFC 4124)
- *Russian Dolls Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering* ed. by F. Le Faucheur (RFC 4127)
- *Maximum Allocation Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering* by F. Le Faucheur & W. Lai (RFC 4125).

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).

Prerequisites

Your network must support the following Cisco IOS 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

Configuration Tasks

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:

	Command	Purpose
Step 1	`Router(config)# ip cef distributed`	Enables CEF—which accelerates the flow of packets through the device.
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 page 3).

	Command	Purpose
Step 4	Router(config)# **router ospf** [or] 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 level-***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:

	Command	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*]]	Enables RSVP on this interface, indicates the Bandwidth Constraints Model to be used (explained on page 3), 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*.

	Command	Purpose
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:

	Command	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...		
this level	**and this item...**	**Use this command**
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 bandwidth on all head routers	**show mpls traffic-eng topology** (described on page 39)
Router	Status of all tunnels currently 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**

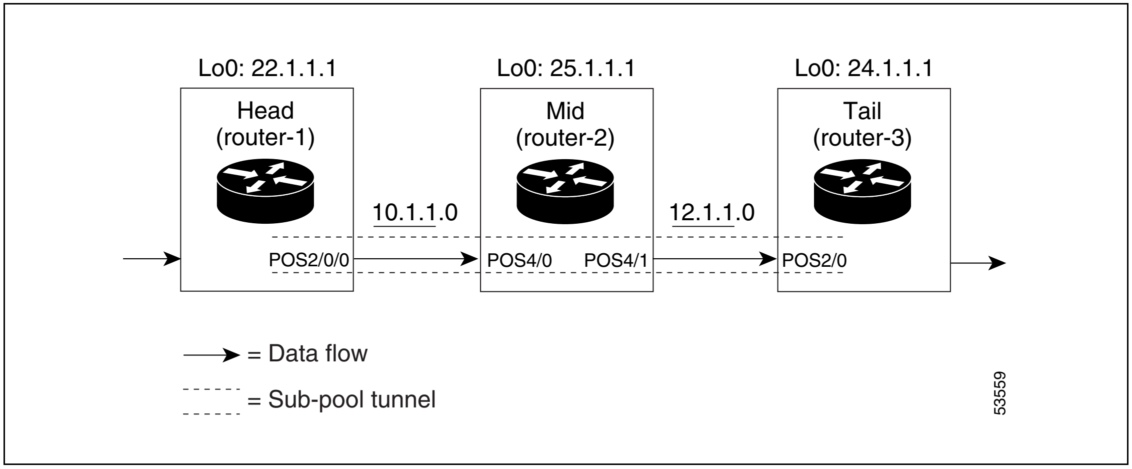
Configuration Examples

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 page 8, 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 page 9 (**ip rsvp bandwidth**), and at the Tunnel Interface Level in Step 4 on page 10 (**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 Figure 1, 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.

Figure 1 Sample Tunnel Topology



Tunnel Head

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-1(config-router)# net
    49.0000.1000.0000.0010.00

    router-1(config-router)# metric-style wide

    router-1(config-router)# is-type level-1
```

```
router ospf 100

redistribute connected


network 10.1.1.0 0.0.0.255 area 0

    network 22.1.1.1 0.0.0.0 area
    0
```

router-1(config-router)# mpls traffic-eng level-1 router-1(config-router)# passive-interface Loopback0	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

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

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

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                     | redistribute connected
    49.0000.1000.0000.0013.00

    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 Configuration: Example

Having configured two bandwidth pools, you now can

- 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:

1. 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.

2. 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.

   You do this 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.

3. 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.

   You do this 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).

4. 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.

   You do this 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.

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.

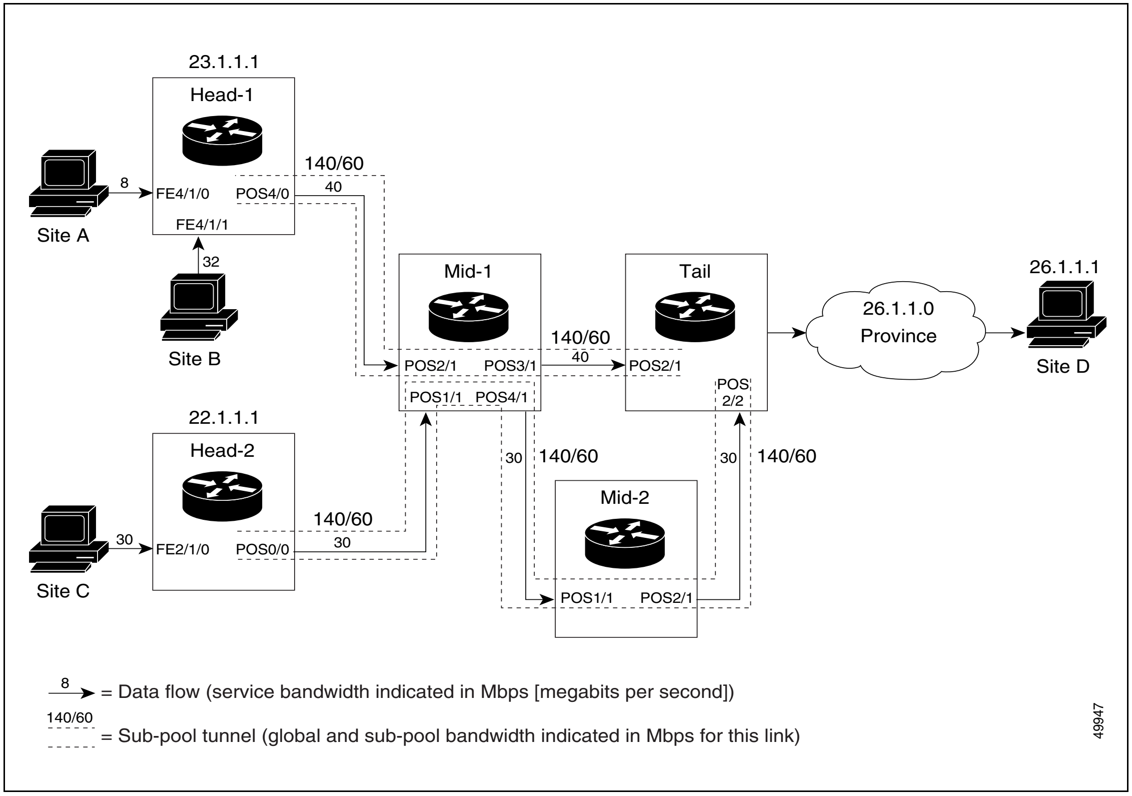
In the second example, there can be many final destinations for the guaranteed bandwidth traffic, including a dynamically changing number of destination prefixes. So the forwarding criteria are specified by Border Gateway Protocol (BGP) policies.

Example with Single Destination Prefix

Figure 2 illustrates a topology 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

Figure 2 Sample Topology for Guaranteed Bandwidth Services to a Single Destination Prefix



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 Figure 2 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                  | redistribute connected
49.0000.1000.0000.0010.00                     |
                                              |
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** router-1(config-router)# **passive-interface Loopback0**	**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)# 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
```

2. 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
```

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

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

c. Packets which exceed this rate are dropped.

d. 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
```

4. 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
```

2. 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
```

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

a. 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

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

c. Packets which exceed this rate are dropped.

d. 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
```

4. 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
```

2. 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
```

3. 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
```

2. 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
```

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

    a. 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

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

    c. Packets which exceed this rate are dropped.

    d. 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
```

4. 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
```

2. 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
```

3. 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

Figure 3 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:

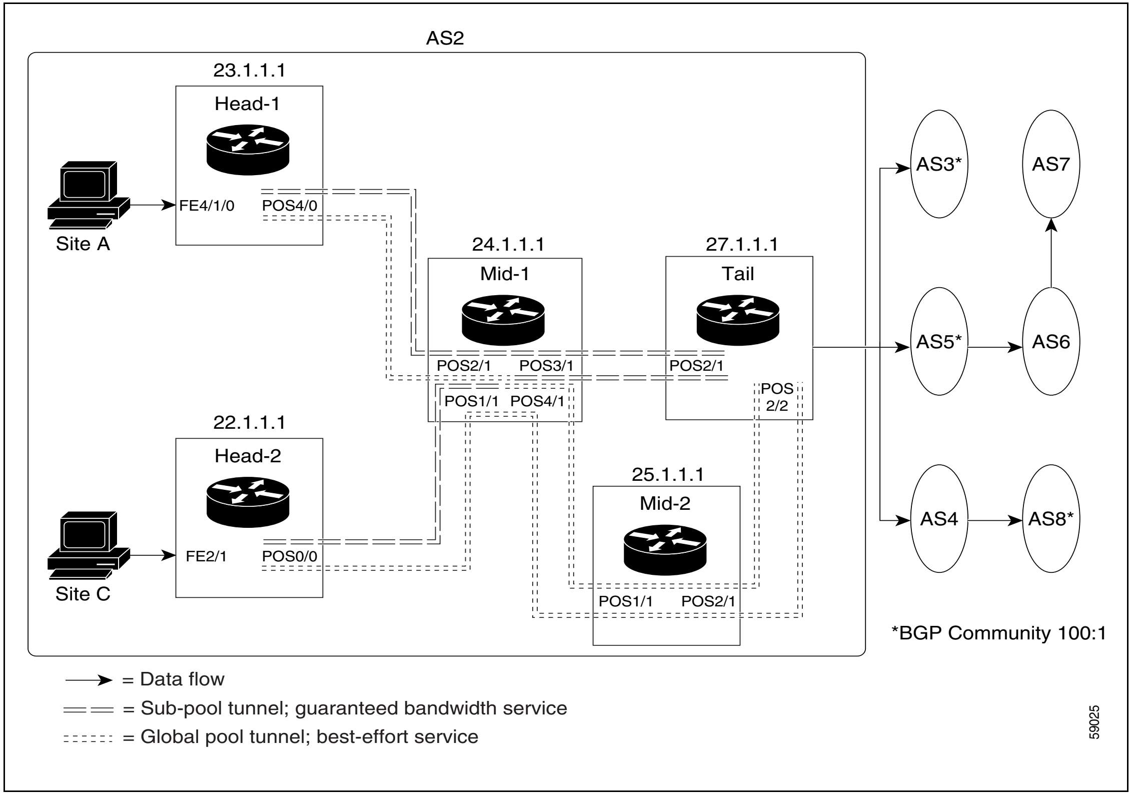
- “Configuring QoS Policy Propagation via Border Gateway Protocol” in the *Cisco IOS Quality of Service Solutions Configuration Guide,* Release 12.1 (http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_c/qcprt1/qcdprop.htm)
- “Configuring BGP” in the *Cisco IOS IP and IP Routing Configuration Guide,* Release 12.1 (http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/ip_c/ipcprt2/1cdbgp.htm)

- "BGP Commands" in the *Cisco IOS IP and IP Routing Command Reference,* Release 12.1 (http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/ip_r/iprprt2/1rdbgp.htm)
- "BGP-Policy Command" in the *Cisco IOS Quality of Service Solutions Command Reference,* Release 12.1 (http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_r/qrdcmd1.htm#xtocid89313)

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.

Figure 3 Sample Topology for Guaranteed Bandwidth Service to Many Destination Prefixes



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:

a. Building a sub-pool MPLS-TE tunnel

b. Configuring DiffServ QoS

c. Configuring QoS Policy Propagation via BGP (QPPB)

d. 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 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 Figure 3 this is FE4/1/0), packets received are rate-limited to:

a. a rate of 30 Mbps

b. a normal burst of 1 MB

c. 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 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                    | redistribute connected
49.0000.1000.0000.0011.00                       |
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       |     mpls traffic-eng area 0
level-1                                         |
```

[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 Figure 3 this is FE2/1), packets received are rate-limited to:

a. a rate of 30 Mbps

b. a normal burst of 1 MB

c. 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                        |redistribute connected
    49.2500.1000.0000.0012.00                           |

    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           |    network 25.1.1.1 0.0.0.0 area
    level-1                                             |    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                      | redistribute connected
    49.0000.2700.0000.0000.00                         |
    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
```

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.

Note See *Internetworking Terms and Acronyms* for terms not included in this glossary.

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.