

This chapter covers the following topics:

- **The Need for Traffic Engineering on the Internet**—Through the deployment of traffic engineering, the traffic flowing across the service provider’s backbone can be optimized, and traffic flows over underutilized paths can be optimized.
- **Unequal-Cost Load Balancing via Metric Manipulation**—This technique allows routers to take advantage of load sharing over multiple unequal-cost paths to a given destination. This can be achieved by manipulating the parameters that determine the routing metrics for protocols such as OSPF, IS-IS, and EIGRP.
- **Advantages of MPLS Traffic Engineering**—This section describes the features provided by MPLS traffic engineering that replicate and expand upon the traffic engineering capabilities of Layer 2 WAN technologies.
- **MPLS Traffic Engineering Elements**—This section describes the various elements of MPLS traffic engineering and their relationships, which together constitute MPLS TE and allow the various elements and functions of traffic engineering to be completely under the control of IP.
- **MPLS Traffic Engineering Configuration**—This section describes the actual configuration steps of MPLS traffic engineering on headend network elements such as MPLS enabled Layer 3 devices.
- **Configuration Case Study of an MPLS Traffic-Engineered Network (IS-IS)**— This case study presents an MPLS traffic-engineered network configured using IS-IS as the Interior Gateway Protocol within the autonomous system.
- **Configuration Case Study of an MPLS Traffic-Engineered Network (OSPF)**— This case study presents an MPLS traffic-engineered network configured using OSPF as the Interior Gateway Protocol within the autonomous system.

MPLS Traffic Engineering

The Need for Traffic Engineering on the Internet

A widespread consensus is that the Internet will transform into a multiservice medium leading to the convergence of voice, video, and data communications. Internet traffic is rising in a geometric progression, with compounded traffic growth. Although the Internet's long-term market performance is difficult to predict, one constant remains—phenomenal growth. Large Internet service providers have responded to the challenge of Internet growth by implementing three complementary initiatives: scalable network architectures, capacity expansion, and traffic engineering (TE).

Internet service providers are ever more challenged to provide the reliability that users have become accustomed to with PSTN and TDM networks. There is also a need to deploy service differentiation in the networks so that ISPs can provide various classes of service at different tariffs. In order to provide such capabilities in the network, the basic traffic-forwarding archetype of the present-day Internet must be enhanced to support traffic engineering. Traffic engineering encompasses many aspects of network performance. These include the provisioning of a guaranteed hard quality of service (QoS), improving the utilization of network resources by distributing traffic evenly across network links, and providing for quick recovery when a node or link fails.

For a service provider to truly and successfully implement commercial Voice over IP (VoIP), a hard QoS with guaranteed delivery of voice packets is required. This can be accomplished by deploying MPLS traffic engineering across the core backbone.

The Internet can be modeled as a collection of autonomous systems communicating with each other using an Exterior Gateway Protocol (EGP). An Interior Gateway Protocol (IGP) is run within the autonomous system to provide any-to-any connectivity. Link-state protocols such as Intermediate System-to-Intermediate System (IS-IS) and Open Shortest Path First (OSPF) typically provide IGP functionality. The EGP currently in use is BGP4. However, Border Gateway Protocol (BGP) is also run within the autonomous system to provide full-mesh Interior Border Gateway Protocol (IBGP) communication between IBGP peers. The IBGP peers might not be directly connected to each other. This is precisely why you need an IGP such as OSPF or IS-IS to provide destination or next-hop routing information for IBGP.

Link-state IGP routing protocols are used to distribute information about all links in the network. Consequently, every IGP router within the autonomous system obtains a complete picture of all the links and routers in the network. Each router then uses this information to compute the shortest path to every possible target subnet in the network using a shortest-path algorithm. The router then builds a forwarding table, associating an address prefix with the next-hop link.

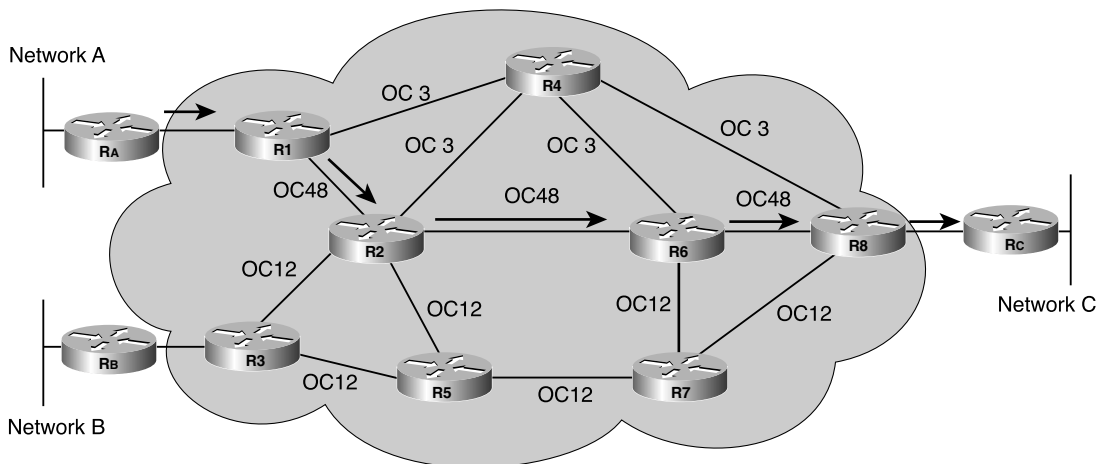
When a packet arrives at a router, the forwarding table is consulted, and the packets are forwarded out on the appropriate link based on the destination IP address. This approach works very well in networks that have a sparse topology. In a network with a densely connected topology, this approach might cause disproportionate network loading. Links that are not on the shortest-path tree remain underutilized despite the presence of heavy traffic loads.

This leads to wasted and underutilized bandwidth on service provider trunks that could otherwise be put to good use.

This issue is currently addressable to some extent by manipulating the link metric used by the routing protocols and forcing unequal-cost load balancing across links. However, these methods do not provide dynamic redundancy and do not consider the characteristics of offered traffic and network capacity constraints when making routing decisions.

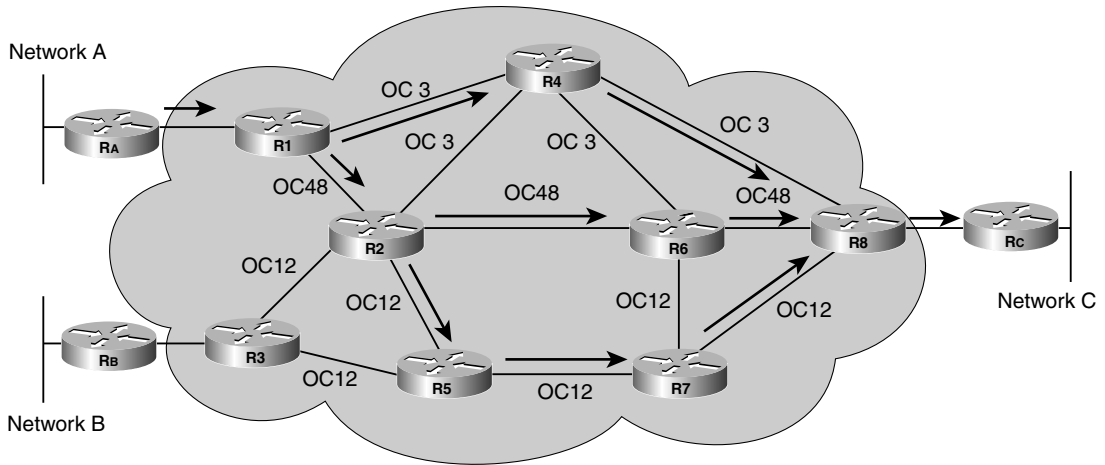
In Figure 7-1, the service provider is running an IGP (for example, OSPF). Based on the cumulative-path cost metric, path R1-R2-R6-R8 is determined to be the best path. All traffic traversing the backbone from R1 to R8 is routed over this path. This leaves the (OC3) links R1-R4-R8 and the mixed high-bandwidth links R1-R2-R5-R7-R8 within the backbone underutilized. Meanwhile, the single OC48 path bears the entire traffic load, making it heavily overutilized.

Figure 7-1 *Underutilized Paths in a Service Provider Backbone*



Through the deployment of traffic engineering, the traffic flowing across the service provider's backbone can be optimized. The routes R1-R4-R8 and R1-R2-R5-R7-R8 can be utilized to load-share the traffic traversing the route between R1 and R8. Figure 7-2 illustrates optimized backbone link utilization for traffic flows between R1 and R8.

Figure 7-2 *Optimized Backbone Link Utilization*



Unequal-Cost Load Balancing via Metric Manipulation

Unequal-cost load balancing is a concept that allows routers to take advantage of load sharing over multiple unequal-cost paths to a given destination. This can be achieved by manipulating the parameters that determine the routing metrics for protocols such as OSPF, IS-IS, and EIGRP.

OSPF Unequal-Cost Load Balancing

Open Shortest Path First (OSPF) uses the cost metric to calculate the best path to a destination network. The path cost is the cumulative sum of the costs assigned to all interfaces that forward traffic along the path to the destination. OSPF calculates the cost based on the link's bandwidth. In general, the path cost in routers is calculated using the formula $10^8/\text{bandwidth}$ (in bps).

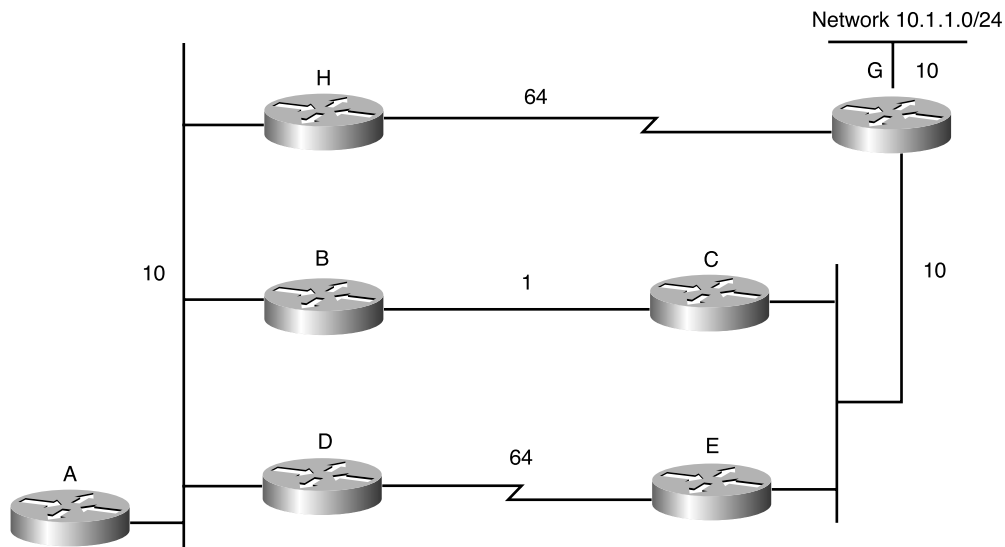
OSPF defaults to equal-cost load balancing. In other words, it load-shares across equal-cost links only. In order to enable OSPF unequal-cost load balancing, you use the **bandwidth** command on the interface. This command might not represent the actual speed of the link, so it can be used to manipulate how data is load-shared over different links with varying

speeds. For OSPF to load-share across links with varying speeds, the **bandwidth** command can be used to set the same value (in bps) across these links. The physical throughput, however, is unchanged, and the command is used only to represent or manipulate the link speed.

For example, in Figure 7-3, there are three ways for Router A to get to Network 10.1.1.0/24:

- A-H-G with a path cost of 84
- A-B-C-G with a path cost of 31
- A-D-E-G with a path cost of 94

Figure 7-3 *OSPF Unequal-Cost Load Balancing*



You can set the **bandwidth** statements on the interfaces such that the path cost for all three paths is equal. The **ip ospf cost cost** command can also be used to change the default cost assigned to a link. This command serves the same purpose as the **bandwidth** command.

NOTE

When changing the path cost using either command, you must be careful that the cost value set conforms to the lowest-speed link. If the value is set according to the highest-speed link, traffic flow will overwhelm the slow links.

EIGRP Unequal-Cost Load Balancing

Most routing protocols support equal-cost-path load balancing. IGRP and EIGRP also support unequal-cost-path load balancing, which is known as *variance*. The **variance** *n* command instructs the router to include routes with a metric smaller than *n* times the minimum metric route for that destination. Traffic is also distributed among the links with respect to the metric.

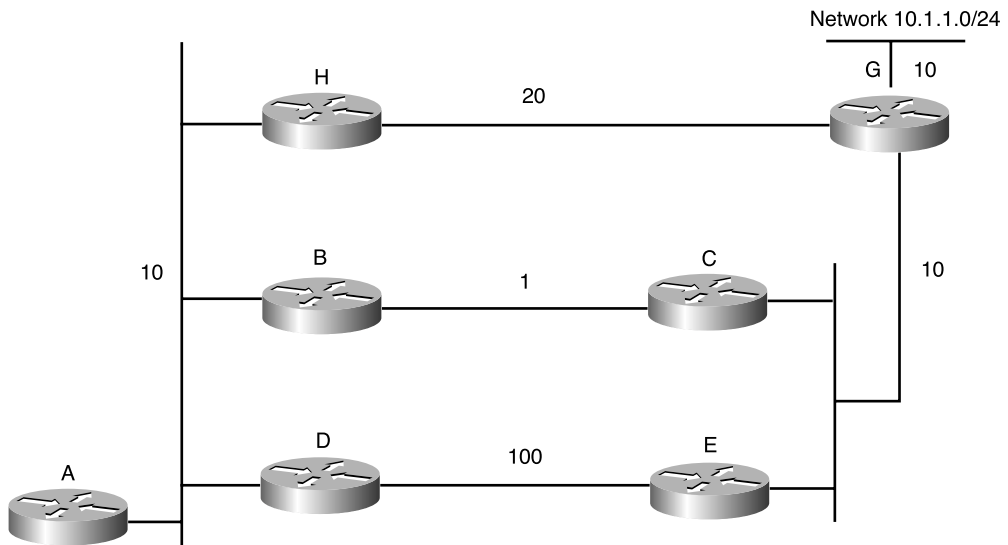
NOTE

According to the EIGRP routing protocol, if a path isn't a feasible successor, it isn't used in load balancing.

Now look at an example. In Figure 7-4, there are three ways to get to Network 10.1.1.0/24:

- A-H-G with a metric of 40
- A-B-C-G with a metric of 31
- A-D-E-G with a metric of 130

Figure 7-4 EIGRP Unequal-Cost Load Balancing



Router A selects the second path, A-B-C-G, with a metric of 31, because 31 is better than 40 or 130. In order to instruct EIGRP to select the path A-H-G as well, configure variance with a multiplier of 2:

```
router eigrp asn
network network-number
variance 2
```

This increases the minimum metric to 62 ($2 \times 31 = 62$). EIGRP includes all the routes that have a metric less than 62 as feasible successors. In the preceding configuration, EIGRP now uses two paths to get to Network 10.1.1.0/24—A-B-C-G and A-H-G—because both paths have a metric less than 62. EIGRP doesn't use path A-D-E-G because it has a metric of 130 and is not a feasible successor.

NOTE EIGRP is currently not supported as an IGP for traffic engineering. The only two link-state IGPs supported by MPLS TE are IS-IS and OSPF. The purpose of the preceding discussion is to demonstrate existing methods of unequal-cost load balancing. EIGRP has a large following in the enterprise community, and the preceding example can be used as a guide to configure unequal-cost load balancing for an EIGRP network.

Metric Manipulation Versus MPLS Traffic Engineering

IP networks exhibit poor efficiency, because the only mechanism for redirecting traffic is to change the link metrics presented to a link-state IGP such as OSPF. However, changing a link's metric can potentially change the path of all packets traversing the link. Also, these methods do not provide dynamic redundancy and do not consider the characteristics of offered traffic and network capacity constraints when making routing decisions.

In an MPLS traffic-engineered network, any Label-Switched Path (LSP) can be dynamically shifted from a congested path to an alternative path. This represents an efficiency improvement over the traditional operational methods for IP networks, because the network managers can run their networks at much higher capacity under normal circumstances, secure in the knowledge that before congestion occurs, some of the traffic can easily be shifted away from the congestion point. Furthermore, network managers can make use of global optimization algorithms that provide a mapping from the traffic demand to the physical links that could not otherwise be achieved using only local optimization. The net result is that a service provider can achieve a much higher degree of link utilization throughout the network, thereby providing services at a lower cost.

MPLS traffic engineering allows service providers to define explicit paths, similar to source routing, across their network and steer traffic over these paths. Redundant explicit paths can be configured, thereby providing a fallback mechanism. Furthermore, a final fallback can be configured. This is typically a dynamic path selected by the IGP. Traffic engineering can also perform Cisco Express Forwarding (CEF)-based unequal-cost load balancing across tunnels. This combination of manual automatic tuning helps realize the goals of capacity planning and helps optimize network utilization on backbone trunks.

Advantages of MPLS Traffic Engineering

MPLS traffic engineering features allow an MPLS backbone to replicate and expand upon the traffic engineering capabilities of Layer 2 ATM and Frame Relay networks. Traffic engineering is essential for service provider and Internet service provider backbones. Both backbones must support a high use of transmission capacity, and the networks must be very resilient so that they can withstand link or node failures. The following are the advantages of MPLS traffic engineering:

- With MPLS, traffic engineering capabilities are integrated into Layer 3, which optimizes the routing of IP traffic given the constraints imposed by backbone capacity and topology.
- It routes IP traffic flows across a network based on the resources the traffic flow requires and the resources available in the network.
- It utilizes *constraint-based routing*, in which the path for a traffic flow is the shortest path that meets the resource requirements or constraints in terms of bandwidth requirements, media requirements, and the traffic flow's priority.
- It dynamically recovers from link or node failures that change the backbone's topology by adapting to a new set of constraints even if several primary paths are precalculated offline.
- It enables unequal-cost load sharing and permits the use of paths other than IGP learned paths.
- It accounts for link bandwidth and for the size of the traffic flow when determining explicit routes across the backbone.
- It replaces the need to manually configure the network devices to set up explicit routes. Instead, you can rely on the MPLS traffic engineering functionality to understand the backbone topology and the automated signaling process.

MPLS Traffic Engineering Elements

The overlay model in which IP is run over an ATM or Frame Relay network results in distinct Layer 2 and Layer 3 networks. The IP network operates over a virtual topology in which every other router is one hop away. This causes difficulties and slows the network's responses to events such as link or node failures. MPLS allows the elements of traffic engineering to be completely under the control of IP. This results in a one-tier network that can offer IP services that now can be achieved only by overlaying a Layer 3 network on a Layer 2 network. This provides a way to achieve the same traffic engineering benefits of the overlay model without needing to run a separate network and without needing a non-scalable full mesh of router interconnects.

MPLS traffic engineering uses Resource Reservation Protocol (RSVP) to automatically establish and maintain a tunnel across the backbone. The path used by a given tunnel at any

point in time is determined based on the tunnel resource requirements and network resources, such as bandwidth. Available resource information is flooded via extensions to a link–state–based IGP such as OSPF or IS-IS.

Tunnel paths are calculated at the tunnel head (source router) based on a fit between required and available resources (constraint-based routing). The IGP automatically routes the traffic into these tunnels. Typically, a packet crossing the MPLS traffic-engineering backbone travels on a single tunnel that connects the ingress point to the egress point.

NOTE

A traffic trunk is an aggregation of microflows that are forwarded along a common path within a service provider’s backbone network from PoP to PoP. These flows normally share a common QoS requirement.

The various elements of MPLS traffic engineering are discussed in the following sections.

LSP Tunnels

LSP tunnels provide the mechanism for steering packets through the MPLS network. They are built using an Integrated Services signaling protocol such as RSVP. LSP tunnels share many of the characteristics of ATM VCs. They are explicitly set up and routed and have a rich set of QoS mechanisms. The RSVP *path* message carries the explicit route to be followed and is used in the interim to allocate resources along the path. The *reservation* message sent in response establishes the label operations and turns the interim allocation into a permanent reservation. When using RSVP, the full QoS offerings of Integrated Services are made available. LSP tunnels are unidirectional. The source router is referred to as the *headend*, and the destination router is the *tail end*. The forward and return paths for an IP flow are independent. Thus, the unidirectional nature of LSP tunnels fits well with traffic engineering of IP traffic.

NOTE

The MPLS header contains 3 experimental bits that are used to represent different Differentiated Services (DiffServ) code points in the future. This results in 2^3 or 8 different DiffServ code points available over a single LSP tunnel.

Distribution of Constraint-Based Routing Information

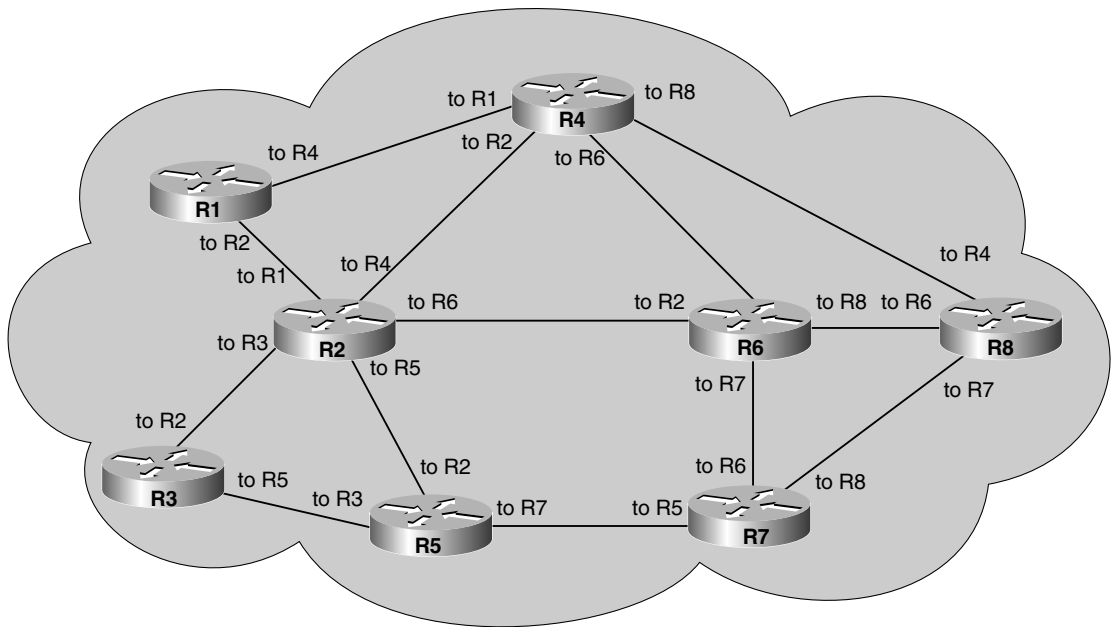
The distribution of constraint-based information must be performed in order to find appropriate paths through the network. LSP traffic-engineered tunnels must be routed with an understanding of the traffic load they need to carry. The constraint information must be

distributed across the MPLS network in a consistent way. The flooding mechanism used by link-state routing protocols such as OSPF and IS-IS can help create an integrated constraint and forwarding database.

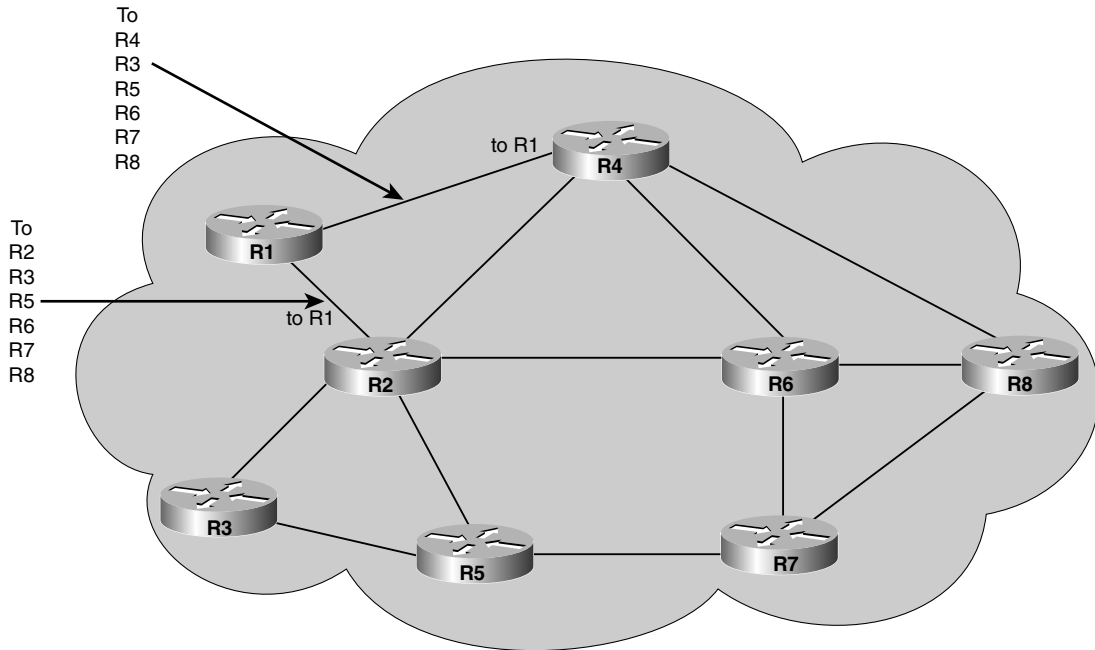
Distance vector (DV) protocols such as RIP are not well-suited for the job due to their limited perception of the network that is confined to just their immediate neighbors. Path determination using DV protocols gets extremely complex, because DV routing tables don't have enough information to calculate alternative paths used by traffic engineering.

This is illustrated in Figures 7-5 and 7-6. Figure 7-5 depicts the network from R1's link state perspective, and Figure 7-6 depicts the network from R1's distance vector perspective.

Figure 7-5 *R1's Link State View of the Network*



OSPF and IS-IS have been extended to carry link constraint information without the need for a separate Layer 2 routing protocol such as ATM Private Network Node Interface (PNNI). MPLS tunnels are not advertised, and updates are not flooded over them. In the overlay model, a single physical link normally carries many virtual circuits (VCs). The failure of a single physical link appears as a failure of multiple links to IP. With MPLS, a single physical link failure appears as a single link failure, thereby reducing flooding and convergence time.

Figure 7-6 R1's Distance Vector View of the Network**NOTE**

For more information on the flooding service from the OSPF IGP, refer to the document on opaque Link State Advertisements (LSAs) for OSPF, draft-katz-yeung-ospf-traffic-04.txt, available at www.ietf.org/internet-drafts/draft-katz-yeung-ospf-traffic-04.txt.

For more information on the flooding service from the IS-IS IGP, refer to the document on new wide TLVs for IS-IS, draft-ietf-isis-traffic-03.txt available at <http://www.ietf.org/internet-drafts/draft-ietf-isis-traffic-03.txt>

Assigning Traffic to Tunnels

The integrated routing feature accomplishes automatic assignment of traffic to tunnels using a modified Shortest Path First (SPF) algorithm. The conventional SPF algorithm runs by iteratively placing contending paths on a *tentative* list, selecting the shortest path from that list, and adding that path and destination node to its forwarding tree. The root node is added to the SPF tree and then adds the one-hop paths to each of its directly connected

neighbors to the tentative list. On each iteration, it adds the current shortest path to its tree and then extends those paths via the links connected to the last node of that path. Routing tables are derived from this shortest-path tree. The routing tables contain ordered sets of destination and first-hop information. If a router does normal hop-by-hop routing, the first hop is a physical interface attached to the router.

Traffic engineering algorithms calculate explicit routes to one or more nodes in the network. These explicit routes are viewed as logical interfaces by the originating router.

These explicit routes are represented by LSPs and are called traffic engineering tunnels (TE tunnels). Link-state IGPs can install routes in the routing table that point to these TE tunnels. These tunnels use explicit routes, and the router that is the headend of the tunnel controls the path taken by a TE tunnel. In the absence of errors, TE tunnels are guaranteed not to loop, but routers must agree on how to use the TE tunnels. Otherwise, traffic might loop through two or more tunnels.

To automatically route traffic onto tunnels, the SPF algorithm is modified as follows: When the endpoint of a tunnel is reached, the next hop to that node is set to the tunnel interface. As the algorithm proceeds, nodes downstream of the tunnel endpoint inherit that tunnel's interface as their next hop. This process continues until the algorithm encounters another node to which it has a tunnel.

This ensures loop-free routing of traffic and provides the same degree of loop prevention provided by link-state routing protocols.

Traffic can also be assigned to LSP tunnels based on BGP next hop or using class of service (CoS) parameters. RSVP defines aggregation over tunnels. LSP tunnels may be used in this way, with the added benefit that they may be routed to where the resources exist if the normal IP route has insufficient resources for the request.

Rerouting

Traffic-engineered networks must be able to respond to changes in network topology and maintain stability. Any link or node failure should not disrupt high-priority network services, especially the higher classes of service. Fast rerouting is a mechanism that minimizes service disruptions for traffic flows affected by an outage, and optimized rerouting reoptimizes traffic flows affected by a change in topology.

Fast Rerouting

In MPLS, splicing and stacking techniques are utilized to enable local repair of LSP tunnels.

Splicing Technique

In this technique, an alternative LSP tunnel is preestablished from the point of protection to the destination via a path that bypasses the downstream network elements being protected. Upon detection of a failure, the forwarding entry for the protected LSP tunnel is updated to use the label and interface of the bypass LSP tunnel.

Stacking Technique

In this technique, a single alternative LSP tunnel, acting as the replacement for the failed link, is created. It bypasses the protected link. The local router maintains a label that represents the bypass tunnel.

NOTE

The alternative LSP tunnel can also be used as a *hop* by another tunnel. Pushing the bypass label onto the stack of labels for packets flowing on the rerouted tunnels does this.

When the protected link fails, all tunnels using that link are updated to use the bypass tunnel. The label forwarding information is updated to first do its normal swap and then push on a label for the bypass tunnel and send the packet out the interface for the bypass tunnel. The label stack is popped at the next-to-last hop of the bypass tunnel. This delivers the labels expected by the next router of the protected LSP tunnel.

Optimized Rerouting

Fast rerouting can result in suboptimal traffic-engineered paths. The key is to dynamically respond to failure as well as to new or restored paths. Thus, when a failure is detected, it is necessary to also notify the headend of the LSP tunnel. The headend can then compute a more optimal path. Traffic can then be diverted to the new LSP tunnel. This can be done without further disruption.

Often missing from Layer 2 networks is a feature called *bridge-and-roll* or *make-before-break*. This is the capability to always set up a new VC while maintaining the current VC. The problem to overcome is this: Suppose the new and existing paths for a tunnel require resources from common links. However, one or more of these links does not have sufficient capacity to admit the second path. The tunnel must first be torn down and then reestablished on the new path. However, if the links can recognize the second path as a replacement for the existing path, the path can be admitted.

RSVP has a reservation style called *shared explicit*. This instructs network elements to use the same capacity to service multiple explicitly named sources. In traffic engineering's use of RSVP, a second path for a tunnel is represented as a different *source* by carrying a path ID as part of the source identification. When a source (the tunnel's headend) wants to

reroute, it sends a path message just as it would for a new tunnel. This message names the same tunnel, but with a new path ID. For links not in common, this appears as a new request. For links that are in common, no new resources need to be allocated. The tail end then sends a reserve message for both paths (senders) using the shared explicit style. The two sender objects are included, and separate label operations are associated with each. As soon as the new path is created, updating the forwarding table diverts traffic. This occurs without service disruption. The old path can then be removed. The presence of the second path message on shared links prevents the cleanup process from removing resources used by the new path.

MPLS Traffic Engineering Configuration

MPLS traffic engineering has certain basic requirements. For example, it is supported by Cisco IOS versions 12.0S, 12.1, 12.1T, and higher service provider IOS images.

The minimum traffic engineering transit configuration tasks are outlined in the following sections.

Configuring a Device to Support MPLS TE Tunnels

To configure a device to support MPLS TE tunnels, do the following:

Step 1 Set up your network with the usual configuration. It is mandatory to set up a loopback interface with a mask of 32 bits. This address is used by the routing protocol for the setup of the MPLS network and TE. This loopback must be in the IGP and must be reachable via the global routing table.

```
Router(config)#interface Loopback n
Router(config-if)#ip address ip-address mask
```

Step 2 Turn on the CEF feature that is necessary for MPLS TE:

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

NOTE

The command **ip cef distributed** can be used on only certain platforms, such as the 7500 and 12000 Gigabit Switch Router, which support distributed processing.

Step 3 Enable the MPLS traffic engineering tunnel feature on the device:

```
Router(config)#mpls traffic-eng tunnels
```

Configuring the Interface(s) to Support RSVP Signaling and IGP Flooding

To configure the interface(s) to support RSVP signaling and IGP flooding, do the following:

- Step 1** Enable the MPLS traffic engineering tunnel feature on all traffic-engineered interfaces to support. This command is needed on both ends of any link an LSP could pass over.

```
Router(config-if)#mpls traffic-eng tunnels
```

- Step 2** Configure all MPLS TE interfaces to support RSVP:

```
Router(config-if)#ip rsvp bandwidth [x-interface-kbps] [y-interface-kbps]
```

The *interface-kbps* argument is optional. It lets you specify the amount of bandwidth in Kbps on the interface to be reserved. The range is 1 to 10,000,000.

x = Maximum reservable bandwidth (default is 75% of available bandwidth)

y = Maximum reservable bandwidth for a single LSP (default is 100% of available bandwidth)

Configuring MPLS Tunnels

MPLS traffic engineering tunnels are unidirectional and are configured at the source router to create an LSP headend. The steps to configure MPLS traffic engineering tunnels are as follows:

- Step 1** Configure a tunnel interface, and enter interface configuration mode:

```
Router(config)#interface Tunnel0
```

- Step 2** Configure the headend to use the IP address of the loopback interface. IOS will not route IP across an interface without an IP address.

```
Router(config-if)#ip unnumbered Loopback0
```

- Step 3** Specify the tunnel mode for MPLS traffic engineering. Other choices for a tunnel encapsulation include GRE and IPSec, which are normally used for VPNs.

```
Router(config-if)#tunnel mode mpls traffic-eng
```

- Step 4** The destination address specifies the tunnel's tail-end router. The address specified for the destination must be the router ID (RID) or loopback interface IP address of the tail-end router.

```
Router(config-if)#tunnel destination IP-address
```

- Step 5** Specify the tunnel's path calculation method. The tunnel has two path setup options—a preferred explicit path and a backup dynamic path. This command configures the tunnel to use a named IP explicit path or a path dynamically calculated from the traffic engineering topology database. A dynamic path is used if an explicit path is currently unavailable.

```
Router(config-if)# tunnel mpls traffic-eng path-option number
{dynamic | explicit {name path-name | path-number}} [lockdown]
```

Explicit path configuration:

```
Router(config)#interface Tunnel0
Router(config-if)#tunnel mpls traffic-eng path-option priority
explicit {idname} IDNAME
```

Dynamic path configuration is as follows. A backup dynamic path can be calculated from the traffic engineering topology database:

```
Router(config-if)tunnel mpls traffic-eng path-option priority dynamic
```

Explicit Path Configuration

A preferred explicit path is set up manually by creating explicit path entries. Each entry indicates a hop to the destination. Each hop specified is a RID or the next-hop interface address of the next-hop router. To enter the subcommand mode for IP explicit paths to create or modify the named path, use the **ip explicit-path** command. An IP explicit path is a list of IP addresses, each representing a node or link in the explicit path. The configuration is as follows:

```
Router(config)# ip explicit-path {name WORD | identifier number} [{enable | disable}]
Router(cfg-ip-expl-path)#next-address next hop RID
Router(cfg-ip-expl-path)#next-address next hop RID
. . .
Router(cfg-ip-expl-path)#next-address next hop RID
Router(cfg-ip-expl-path)#exit
```

Configuring an MPLS TE Tunnel for IGP Use

To configure an MPLS traffic engineering tunnel that an IGP can use, perform these steps in interface configuration mode. If these steps are not executed, the tunnel will come up but will not be used.

- Step 1** Configure an interface type, and enter interface configuration mode:

```
Router(config-if)# interface tunnel0
```

- Step 2** Announce the tunnel tail-end reachability to the Routing Information Base (RIB). This causes the IGP to use the tunnel in its enhanced SPF calculation.

```
Router(config-if)#tunnel mpls traffic-eng autoroute announce
```

Configuring IS-IS for MPLS TE

Recently, new extensions have been designed and implemented for the IS-IS routing protocol. These extensions serve multiple purposes. One goal is to remove the 6-bit limit on link metrics. A second goal is to allow for interarea IP routes. A third goal is to allow IS-IS to carry different kinds of information for the purpose of traffic engineering. In the future, more extensions might be needed. To serve these purposes, two new TLVs have been defined. (TLV stands for type, length, and value object.) The first new TLV (TLV 22) describes links (or, rather, adjacencies). The second new TLV (TLV 135) describes reachable IP prefixes. Both new TLVs have a fixed-length part followed by optional sub-TLVs. The metric space in these new TLVs has been enhanced from 6 bits to 24 or 32 bits. The sub-TLVs allow you to add new properties to links and prefixes. Traffic engineering is the first technology to make use of this ability to describe new properties of a link.

IS-IS Migration Solution 1

One solution when you are migrating from old-style TLVs toward new-style TLVs is to advertise the same information twice—once in old-style TLVs and once in new-style TLVs. This ensures that all routers have the opportunity to understand what is advertised. However, this approach has two obvious drawbacks:

- **The size of the LSPs**—During transition, the LSPs grow roughly two times in size. This might be a problem in networks where the LSPDB is large. An LSPDB can be large because there are many routers and thus LSPs. Or, the LSPs are large because of many neighbors or IP prefixes per router. A router that advertises a lot of information causes the LSPs to be fragmented. A large network in transition pushes the limits of LSP flooding and SPF scaling. During the transition, you can expect some extra network instability. During this time, you especially do not want to test how far you can push an implementation. There is also the possibility that the traffic engineering extensions might cause LSPs to be reflooded more often. For a large network, this solution could produce unpredictable results.
- **The problem of ambiguity**—If you choose this solution, you might get an ambiguous answer to a question such as this: What should a router do if it encounters different information in the old-style TLVs and new-style TLVs?

This problem can largely be solved easily by using all information in old-style and new-style TLVs in an LSP. The router uses the adjacency with the lowest link metric if an adjacency is advertised more than once. The main benefit is that network administrators can use new-style TLVs before all routers in the network can understand them.

IS-IS Migration Solution 1 Transition Steps

Here are some steps you can follow when transitioning from using IS-IS with old-style TLVs to new-style TLVs:

- Step 1** Advertise and use only old-style TLVs if all routers run old software.
- Step 2** Upgrade some routers to newer software.
- Step 3** Configure some routers with new software to advertise both old-style and new-style TLVs. They accept both styles of TLVs. Configure other routers (with old software) to keep advertising and using only old-style TLVs.
- Step 4** Test traffic engineering in parts of the network. However, wider metrics cannot be used yet.
- Step 5** If the whole network needs to migrate, upgrade and configure all remaining routers to advertise and accept both styles of TLVs.
- Step 6** Configure all routers to advertise and accept only new-style TLVs.
- Step 7** Configure metrics larger than 63.

IS-IS Migration Solution 2

Routers advertise only one style of TLV at the same time but can understand both types of TLVs during migration. One benefit is that LSPs stay roughly the same size during migration. Another benefit is that there is no ambiguity between the same information advertised twice inside one LSP.

The drawback is that all routers must understand the new-style TLVs before any router can start advertising new-style TLVs. So, this transition scheme is useful when transitioning the whole network (or a whole area) to use wider metrics. It does not help solve the second problem, in which network administrators want to use the new-style TLVs for traffic engineering while some routers can still understand only old-style TLVs.

IS-IS Migration Solution 2 Transition Steps

Here are some steps you can follow when transitioning from using IS-IS with old-style TLVs to a combination of old- and new-style TLVs:

- Step 1** Advertise and use only old-style TLVs if all routers run old software.
- Step 2** Upgrade all routers to newer software.
- Step 3** Configure all routers one-by-one to advertise old-style TLVs and to accept both styles of TLVs.
- Step 4** Configure all routers one-by-one to advertise new-style TLVs and to accept both styles of TLVs.
- Step 5** Configure all routers one-by-one to advertise and accept only new-style TLVs.
- Step 6** Configure metrics larger than 63.

Configuring IS-IS for MPLS TE Within the AS IS-IS routing must be properly configured for IP within the autonomous system as the IGP using a proper IP architecture. Do the following to accomplish this:

- Step 1** Enable IS-IS routing, and specify an IS-IS process for IP, which places you in router configuration mode:

```
Router(config)#router isis
```

- Step 2** Turn on MPLS traffic engineering for IS-IS level 1 or 2:

```
Router(config-router)#mpls traffic-eng level [1 | 2]
```

NOTE

Currently, MPLS traffic engineering does not support level 2 IS-IS. Most ISP backbones are either all level 1 or all level 2.

- Step 3** Specify the traffic engineering router identifier for the node to be the IP address associated with interface loopback0:

```
Router(config-router)#mpls traffic-eng router-id loop0
```

- Step 4** Configure the router to generate and accept only new-style TLVs:

```
Router(config-router)#metric-style wide
```

Configuring OSPF for MPLS TE

To configure OSPF for MPLS TE, follow these steps:

NOTE OSPF uses type 10 LSAs (also called opaque LSAs).

Step 1 Enable the OSPF process on the router and specify the Process ID (PID), which places you in router configuration mode:

```
Router(config)#router ospf pid
```

Step 2 Configuring OSPF for MPLS requires traffic engineering to be configured in an area:

```
Router(config-router)#mpls traffic-eng area area
```

NOTE Currently, OSPF supports MPLS traffic engineering in only a single area—typically, the backbone or Area 0.

Step 3 Explicitly configure the RID. The IP address of the loopback interface is used as the RID.

```
Router(config-router)#mpls traffic-eng area router-id loop0
```

Configuring MPLS Tunnel Unequal-Cost Load Balancing

Unequal-cost load balancing can be configured between two or more MPLS traffic engineering tunnels with the same destination tail end. The bandwidth parameter used for load balancing is specified in kilobits per second. The default bandwidth is 0.

```
Router(config)#interface Tunnel0
Router(config-if)#tunnel destination destination IP address
Router(config-if)#tunnel mpls traffic-eng bandwidth x

Router(config)#interface Tunnel1
Router(config-if)#tunnel destination destination IP address
Router(config-if)#tunnel mpls traffic-eng bandwidth y
```

Verifying MPLS Traffic Engineering Operation

The following steps show you how to verify MPLS traffic engineering operation:

Step 1 Display information about the MPLS TE tunnels using the **show mpls traffic-eng tunnel** command:

```
show mpls traffic-eng tunnel [tunnel_interface | destination address |
source-id {ip-address | 0-MAX | name name role
```

```
{all | head | middle | tail | remote} | {up | down}}] [brief]
```

The following example shows a sample output from the **show mpls traffic-eng tunnel brief** command:

```
R1#show mpls traffic-eng tunnel brief

Signaling Summary:
  LSP Tunnels Process:      running
  RSVP Process:            running
  Forwarding:              enabled
  Periodic reoptimization:  every 180 seconds, next in 108 seconds
TUNNEL NAME                DESTINATION    STATUS    STATE
R1_t0                      10.10.10.8    up/up    up/up
R1_t1                      10.10.10.8    up/up    up/up
...
Displayed 2 (of 2) heads, 0 (of 0) midpoints, 1 (of 1) tails
```

The detailed configuration of any tunnel can be seen using the following:

```
R1#show mpls traffic-eng tunnels name R1_t0

Name: R1_t0                (Tunnel0) Destination: 10.10.10.8
Status:
Admin: up                 Oper: up                 Path: valid              Signaling: connected
      path option 1, type explicit low (Basis for Setup, path weight 40)
Config Parameters:
Bandwidth: 120000 kbps Priority: 2 2 Affinity: 0x0/0xFFFF
AutoRoute: enabled       LockDown: disabled
InLabel : -
OutLabel : atm4/0/0.1, 17
RSVP Signaling Info:
Src 10.10.10.1, Dst 10.10.10.8, Tun_Id 0, Tun_Instance 1601
RSVP Path Info:
My Address: 10.10.10.1
Explicit Route: 10.10.12.2 10.10.25.2 10.10.57.2 10.10.78.2
Record Route: NONE
Tspec:av rate=120000 kbits, burst=8000 bytes,peak rate=120000 kbits
RSVP Resv Info:
Record Route: NONE
Fspec: av rate=120000 kbits, burst=8000bytes, peak rate=84974967 kbits
History:
Current LSP:
Uptime: 3 hours, 33 minutes
Selection: reoptimization
Prior LSP:
ID: path option 1 [1600]
Removal Trigger: configuration changed
```

In the preceding case, the path is explicit and specified in the RSVP message. (The field that carries the path is also known as the Explicit Route Object [ERO].) If this path cannot be followed, the MPLS TE engine uses the next path option, which can be another explicit route or a dynamic route.

Step 2 Display RSVP information about the interfaces using the **show ip rsvp interface** command:

```
show ip rsvp interface
```

In the following output, on R4, four reservations are made, each of 30000K:

```
R4#show ip rsvp interface
```

interface	allocated	i/f max	flow max	pct	UDP	IP	UDP_IP	UDP M/C
atm4/0/0	0M	0M	0M	0	0	0	0	0
atm4/0/0.1	30000K	30000K	30000K	30	0	1	0	0
atm4/0/1	0M	0M	0M	0	0	0	0	0
atm4/0/1.1	30000K	30000K	30000K	30	0	1	0	0
atm4/0/2	0M	0M	0M	0	0	0	0	0
atm4/0/2.1	30000K	30000K	30000K	30	0	1	0	0
atm4/0/3	0M	0M	0M	0	0	0	0	0
atm4/0/3.1	30000K	30000K	30000K	30	0	1	0	0

Step 3 Display the TE path that will be used for a particular destination (and a particular bandwidth) without creating a tunnel:

```
show mpls traffic-eng topology path destination dest-ip-address
bandwidth bandwidth-in-kbps
```

The following is an example:

```
R1#show mpls traffic-eng topology path destination 10.10.10.8 bandwidth 200000
```

Query Parameters:

```
Destination: 10.10.10.8
Bandwidth: 200000
Priorities: 0 (setup), 0 (hold)
Affinity: 0x0 (value), 0xFFFFFFFF (mask)
```

Query Results:

```
Min Bandwidth Along Path: 622000 (kbps)
Max Bandwidth Along Path: 2500000 (kbps)
Hop 0: 10.10.12.1 : affinity 00000000, bandwidth 2500000(kbps)
Hop 1: 10.10.25.1 : affinity 00000000, bandwidth 622000 (kbps)
Hop 2: 10.10.57.1 : affinity 00000000, bandwidth 622000 (kbps)
Hop 2: 10.10.78.1 : affinity 00000000, bandwidth 620000 (kbps)
Hop 3: 10.10.10.8
```

Step 4 To display a log of 20 entries of MPLS traffic engineering IS-IS adjacency changes, use the **show isis mpls traffic-eng adjacency-log** EXEC command:

```
show isis mpls traffic-eng adjacency-log
```

Example 7-1 Step 4 – Example 1

```
R1#show isis mpls traffic-eng adjacency-log

IS-IS RRR log
When      Neighbor ID      IP Address      Interface Status Level
04:52:52  0000.0024.0004.02  0.0.0.0        Et0/2      Up      level-1
04:52:50  0000.0026.0001.00  170.1.1.2      P01/0/0    Up      level-1
04:52:37  0000.0024.0004.02  0.0.0.0        Et0/2      Up      level-1
```

Step 5 To display RSVP terminal point information for receivers or senders, use the **show ip rsvp host** EXEC command:

```
show ip rsvp host {host {receivers | senders} | installed | interface |
neighbor | request | reservation | sender}
```

Example 7-2 Step 5 – Example 1

```
R1# show ip rsvp host receivers
To          From          Pro DPort Sport Next Hop      I/F  Fi Serv BPS Bytes
10.0.0.11   10.1.0.4      0  10011 1          I/F  SE LOAD 100K 1K
```

Step 6 To display the last flooded record from MPLS traffic engineering, use the **show isis mpls traffic-eng advertisements** EXEC command:

```
show isis mpls traffic-eng advertisements
```

Example 7-3 Step 6 – Example 1

```
R1#show isis mpls traffic-eng advertisements

System ID:dtp-5.00
Router ID:5.5.5.5
Link Count:1
Link[1]
Neighbor System ID:dtp-5.01 (broadcast link)
Interface IP address:172.21.39.5
Neighbor IP Address:0.0.0.0
Admin. Weight:10
Physical BW:10000000 bits/sec
Reservable BW:1166000 bits/sec
BW unreserved[0]:1166000 bits/sec, BW unreserved[1]:1166000 bits/sec
BW unreserved[2]:1166000 bits/sec, BW unreserved[3]:1166000 bits/sec
BW unreserved[4]:1166000 bits/sec, BW unreserved[5]:1166000 bits/sec
BW unreserved[6]:1166000 bits/sec, BW unreserved[7]:1153000 bits/sec
Affinity Bits:0x00000000
```

Step 7 To show summary information about tunnels, use the **show mpls traffic-eng tunnel summary** command:

```
show mpls traffic-eng tunnel summary
```

Example 7-4 Step 7 – Example 1

```
R1# show mpls traffic-eng tunnel summary

Signalling Summary:
  LSP Tunnels Process:      running
  RSVP Process:            running
  Forwarding:              enabled
  Head: 1 interfaces, 1 active signalling attempts, 1 established
        1 activations, 0 deactivations
  Midpoints: 0, Tails: 0
  Periodic reoptimization:  every 3600 seconds, next in 3436 seconds
```

Step 8 To show the MPLS traffic engineering global topology as currently known at this node, use the **show mpls traffic-eng topology** privileged EXEC command:

```
show mpls traffic-eng topology [A.B.C.D | igp-id {isis nsapaddr |
ospf A.B.C.D}] [brief]
```

Example 7-5 Step 8 – Example 1

```
R1#show mpls traffic-eng topology

My_System_id: 0000.0025.0003.00

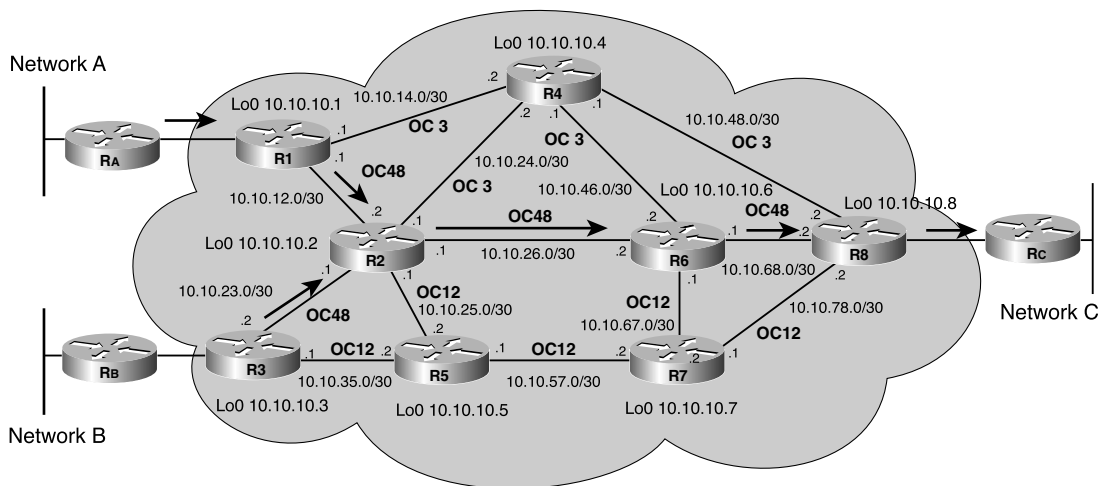
IGP Id: 0000.0024.0004.00, MPLS TE Id:24.4.4.4 Router Node
link[0 ]:Intf Address: 150.1.1.4
      Nbr IGP Id: 0000.0024.0004.02,
      admin_weight:10, affinity_bits:0x0
      max_link_bw:10000 max_link_reservable: 10000
      allocated   reservable   allocated   reservable
      -----
      bw[0]: 0           10000      bw[1]: 0           10000
      bw[2]: 0           10000      bw[3]: 0           10000
      bw[4]: 0           10000      bw[5]: 0           10000
      bw[6]: 0           10000      bw[7]: 0           10000
```

Configuration Case Study of an MPLS Traffic-Engineered Network (IS-IS)

Consider a service provider that has the network topology shown in Figure 7-7. In this example, the network is running over an ATM backbone, and the link-state routing protocol being used is IS-IS. The links between R1-R2-R6-R8 are OC48 (2.5 Gbps). The rest of the links within the service provider cloud are OC3 (155 Mbps) and OC12 (622 Mbps). Based

on the link-state routing algorithm, traffic traversing from Network A to Network C is routed across the best path determined on the basis of an IS-IS metric. Therefore, the path across R1-R2-R6-R8 is selected for routing this traffic because it has the lowest cumulative path cost. Similarly, traffic between Network B and Network C is routed through Routers R3-R2-R6-R8, leaving the other links within the cloud underutilized. The underutilized paths in the backbone are R1-R4-R8 and R1-R2-R5-R7-R8 for traffic flowing between Network A and Network C, and the links R3-R5-R7-R8 and R3-R2-R4-R8 are underutilized for traffic flowing between Network B and Network C.

Figure 7-7 MPLS TE Case Study Topology and Traffic Flow R1-R3 to R8

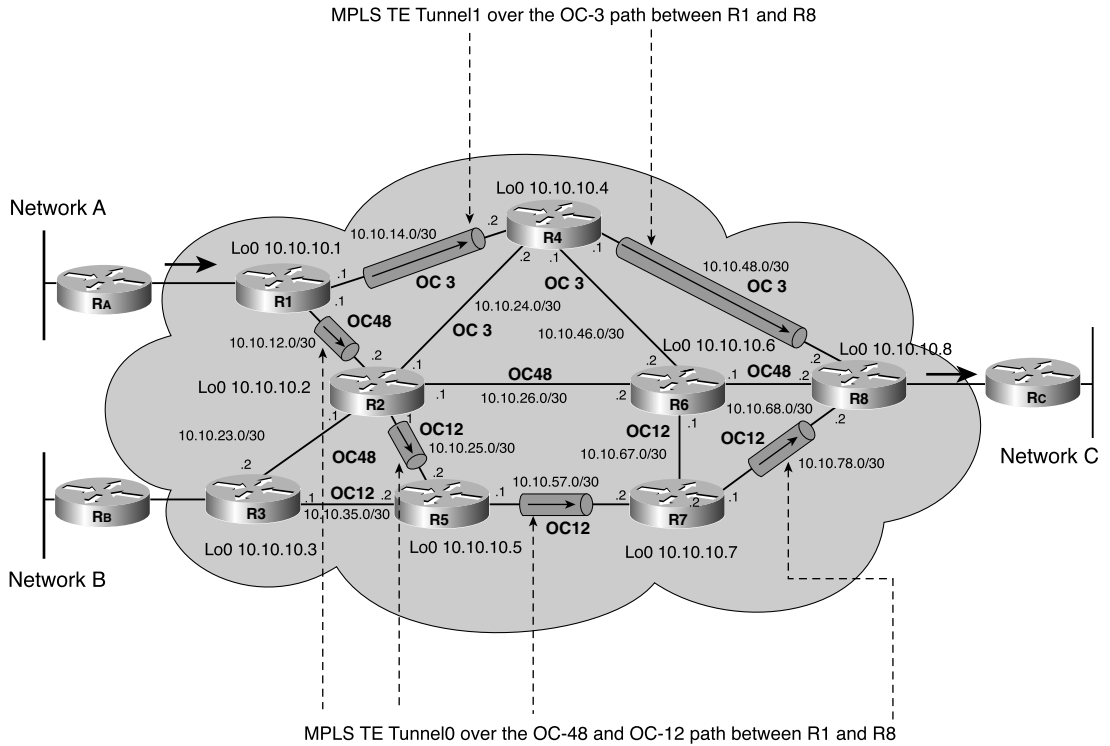


Implementing MPLS traffic engineering can optimize network resource utilization and evenly spread traffic across the underutilized links.

R1 Traffic Engineering Policy

R_A uses the IGP selected path R1-R2-R6-R8 by default in order to access R_C. As shown in Figure 7-8, MPLS traffic engineering tunnels Tunnel0 and Tunnel1 steer traffic through the underutilized paths R1-R2-R5-R7-R8 and R1-R4-R8, respectively. Tunnel0 has been configured to utilize R1-R2-R5-R7-R8 (the OC12 path) as its first path (in order of priority) and R1-R4-R8 (the OC3 path) as its second path (in order of priority). The dynamic path is the fallback path if the first and second paths are unavailable due to link or node failure. The dynamic path is normally the IGP derived path. In this case study, the IGP used is IS-IS.

Figure 7-8 R1 to R8 Traffic Engineering Tunnels



Tunnel1 has been configured to utilize R1-R4-R8 (the OC3 path) as its first path (in order of priority) and R1-R2-R5-R7-R8 (the OC12 path) as its second path (in order of priority). It uses the dynamic path in the same way as Tunnel0.

The network has also been traffic-engineered to load-balance across Tunnel0 and Tunnel1. The load balancing is achieved by configuring bandwidth statements within each tunnel interface. The ratio of these values is used by CEF to make load-balancing decisions.

R1 Configuration (IS-IS)

The configuration of R1 is as follows:

```

!
hostname R1
!
ip cef
mpls traffic-eng tunnels
!
    
```

continues

```
interface Loopback0
  ip address 10.10.10.1 255.255.255.255
  ip router isis
  !
interface Tunnel0
  ip unnumbered Loopback0
  tunnel destination 10.10.10.8
  tunnel mode mpls traffic-eng
  tunnel mpls traffic-eng autoroute announce
  tunnel mpls traffic-eng priority 1 1
  tunnel mpls traffic-eng bandwidth 120000
  tunnel mpls traffic-eng path-option 10 explicit name r1r8_oc12path
  tunnel mpls traffic-eng path-option 20 explicit name r1r8_oc3path
  tunnel mpls traffic-eng path-option 30 dynamic
  !
interface Tunnel1
  ip unnumbered Loopback0
  tunnel destination 10.10.10.8
  tunnel mode mpls traffic-eng
  tunnel mpls traffic-eng autoroute announce
  tunnel mpls traffic-eng priority 2 2
  tunnel mpls traffic-eng bandwidth 30000
  tunnel mpls traffic-eng path-option 10 explicit name r1r8_oc3path
  tunnel mpls traffic-eng path-option 20 explicit name r1r8_oc12path
  tunnel mpls traffic-eng path-option 30 dynamic
  !
interface atm4/0/0
  no ip address
  no ip directed broadcast
  no atm ilmi-keepalive
  !
interface atm4/0/0.1 point-to-point
  description OC48 to R2
  bandwidth 2500000
  ip address 10.10.12.1 255.255.255.252
  ip router isis
  tag-switching ip
  mpls traffic-eng tunnels
  pvc 2/5
  encapsulation aal5snap
  ip rsvp bandwidth 500000 500000
  !
interface atm4/0/1
  no ip address
  no ip directed broadcast
  no atm ilmi-keepalive
  !
interface atm4/0/1.1 point-to-point
  description OC3 to R4
  bandwidth 155000
  ip address 10.10.14.1 255.255.255.252
  ip router isis
```

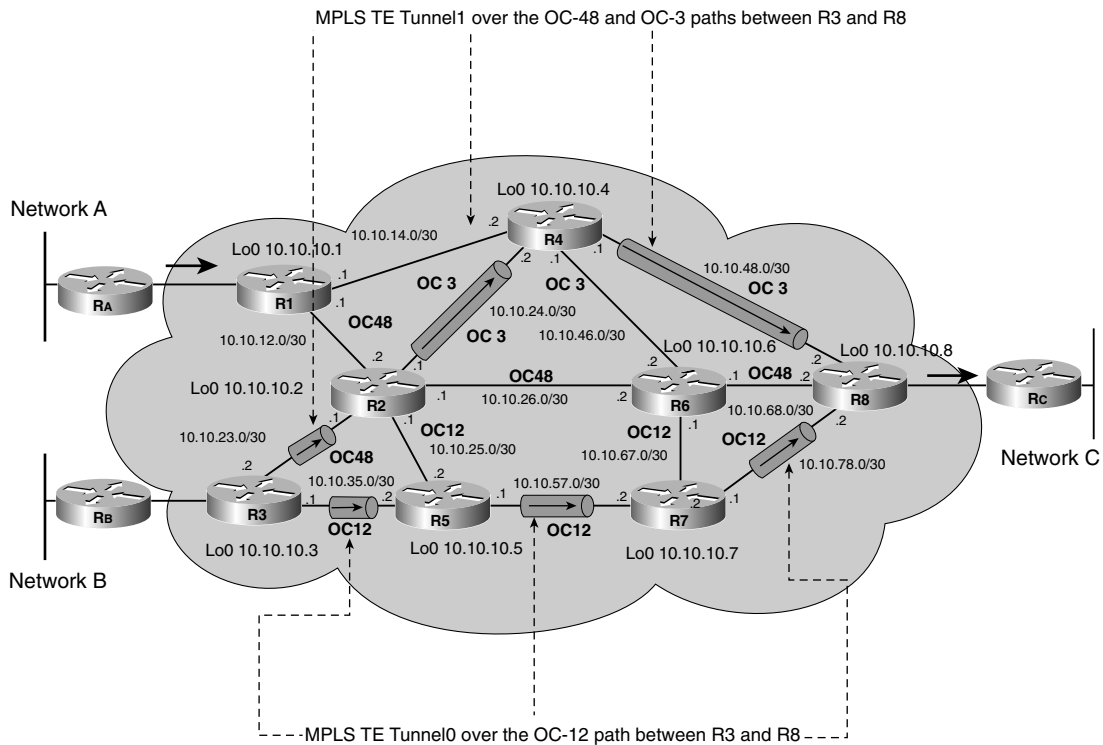
```
tag-switching ip
mpls traffic-eng tunnels
pvc 3/5
encapsulation aal5snap
ip rsvp bandwidth 30000 30000
!
router isis
net 49.0001.0000.0000.0001.00
is-type level-1
metric-style wide
mpls traffic-eng router-id Loopback0
mpls traffic-eng level-1
!
ip classless
!
ip explicit-path name oc12path enable
next-address 10.10.12.2
next-address 10.10.25.2
next-address 10.10.57.2
next-address 10.10.78.2
!
ip explicit-path name oc3path enable
next-address 10.10.14.2
next-address 10.10.48.2
!
end
```

R3 Traffic Engineering Policy

R_B uses the IGP selected path R3-R2-R6-R8 by default in order to access R_C. In Figure 7-9, MPLS traffic engineering tunnels Tunnel0 and Tunnel1 steer traffic through the underutilized paths R3-R5-R7-R8 and R3-R2-R4-R8, respectively. Tunnel0 has been configured to utilize R3-R5-R7-R8 (the OC12 path) as its first path (in order of priority) and R3-R2-R4-R8 (the OC3 path) as its second path (in order of priority). The dynamic path is the fallback path if the first and second paths are unavailable due to link or node failure. The dynamic path is normally the IGP derived path. In this case study, the IGP used is IS-IS.

Tunnel1 has been configured to utilize R3-R2-R4-R8 (the OC3 path) as its first path (in order of priority) and R3-R5-R7-R8 (the OC12 path) as its second path (in order of priority). It uses the dynamic path in the same way as Tunnel0.

The network has also been traffic-engineered to load-balance across Tunnel0 and Tunnel1. The load balancing is achieved by configuring bandwidth statements within each tunnel interface. The ratio of these values is used by CEF to make load-balancing decisions.

Figure 7-9 R3 to R8 Traffic Engineering Tunnels

R3 Configuration (IS-IS)

The configuration of R3 is as follows:

```

!
hostname R3
!
ip cef
mpls traffic-eng tunnels
!
interface Loopback0
 ip address 10.10.10.3 255.255.255.255
 ip router isis
!
interface Tunnel0
 ip unnumbered Loopback0
 tunnel destination 10.10.10.8

```

```
tunnel mode mpls traffic-eng
tunnel mpls traffic-eng autoroute announce
tunnel mpls traffic-eng priority 1 1
tunnel mpls traffic-eng bandwidth 120000
tunnel mpls traffic-eng path-option 10 explicit name r3r8_oc12path
tunnel mpls traffic-eng path-option 20 explicit name r3r8_oc3path
tunnel mpls traffic-eng path-option 30 dynamic
!
interface Tunnel1
ip unnumbered Loopback0
tunnel destination 10.10.10.8
tunnel mode mpls traffic-eng
tunnel mpls traffic-eng autoroute announce
tunnel mpls traffic-eng priority 2 2
tunnel mpls traffic-eng bandwidth 30000
tunnel mpls traffic-eng path-option 10 explicit name r3r8_oc3path
tunnel mpls traffic-eng path-option 20 explicit name r3r8_oc12path
tunnel mpls traffic-eng path-option 30 dynamic
!
interface atm4/0/0
no ip address
no ip directed broadcast
no atm ilmi-keepalive
!
interface atm4/0/0.1 point-to-point
description OC48 to R2
bandwidth 2500000
ip address 10.10.23.2 255.255.255.252
ip router isis
tag-switching ip
mpls traffic-eng tunnels
pvc 4/6
encapsulation aal5snap
ip rsvp bandwidth 500000 500000
!
interface atm4/0/1
no ip address
no ip directed broadcast
no atm ilmi-keepalive
!
interface atm4/0/1.1 point-to-point
description OC12 to R5
bandwidth 622000
ip address 10.10.35.1 255.255.255.252
ip router isis
tag-switching ip
mpls traffic-eng tunnels
pvc 5/8
encapsulation aal5snap
ip rsvp bandwidth 120000 120000
!
router isis
```

continues

```

net 49.0003.0000.0000.0003.00
is-type level-1
metric-style wide
mpls traffic-eng router-id Loopback0
mpls traffic-eng level-1
!
ip classless
!
ip explicit-path name r3r8_oc12path enable
next-address 10.10.35.2
next-address 10.10.57.2
next-address 10.10.78.2
!
ip explicit-path name r3r8_oc3path enable
next-address 10.10.23.1
next-address 10.10.24.2
next-address 10.10.48.2
end

```

R8 Traffic Engineering Policy

Figure 7-10 shows the default traffic flows between Network C and Network A or B. R_C uses the IGP selected path R8-R6-R2-R1 by default in order to access R_A and uses R8-R6-R2-R3 to access R_B .

In Figure 7-11, MPLS traffic engineering tunnels Tunnel0 and Tunnel1 steer traffic between R_C and R_A through the underutilized paths R8-R7-R5-R2-R1 and R8-R4-R1, respectively. Tunnel0 has been configured to utilize R8-R7-R5-R2-R1 (the OC12 path) as its first path (in order of priority) and R8-R4-R1 (the OC3 path) as its second path (in order of priority). The dynamic path is the fallback path if the first and second paths are unavailable due to link or node failure.

Tunnel1 has been configured to utilize R8-R4-R1 (the OC3 path) as its first path (in order of priority) and R8-R7-R5-R2-R1 (the OC12 path) as its second path (in order of priority). The dynamic path is the fallback path if the first and second paths are unavailable due to link or node failure. The dynamic path is normally the IGP derived path. In this case study, the IGP used is IS-IS.

Similarly, as shown in Figure 7-12, Tunnel2 and Tunnel3 steer traffic between R_C and R_B through the underutilized paths R8-R7-R5-R3 and R8-R4-R2-R3, respectively. Tunnel2 has been configured to utilize R8-R7-R5-R3 (the OC12 path) as the first path (in order of priority) and R8-R4-R2-R3 (the OC3 path) as the second path (in order of priority). The dynamic path is the fallback path if the first and second paths are unavailable due to link or node failure.

Figure 7-10 MPLS TE Case Study Topology and Traffic Flow R8 to R1-R3

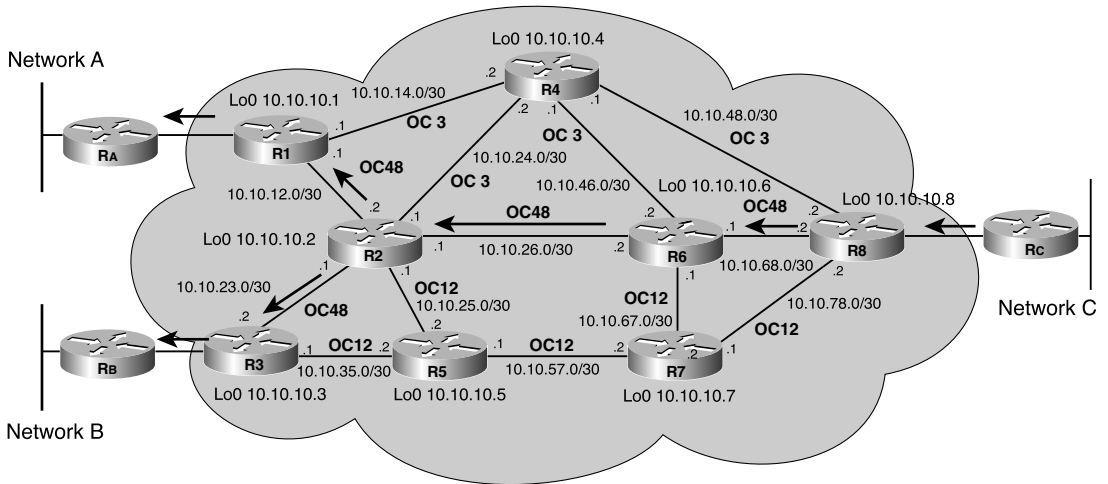


Figure 7-11 R8 to R1 Traffic Engineering Tunnels

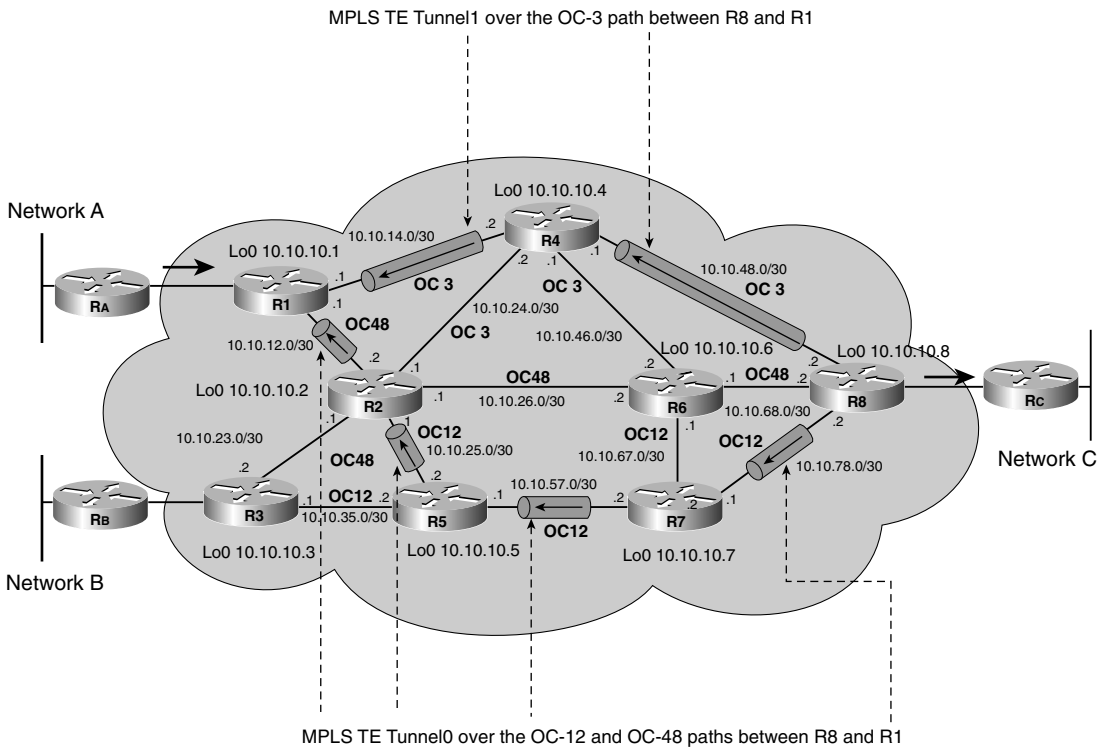
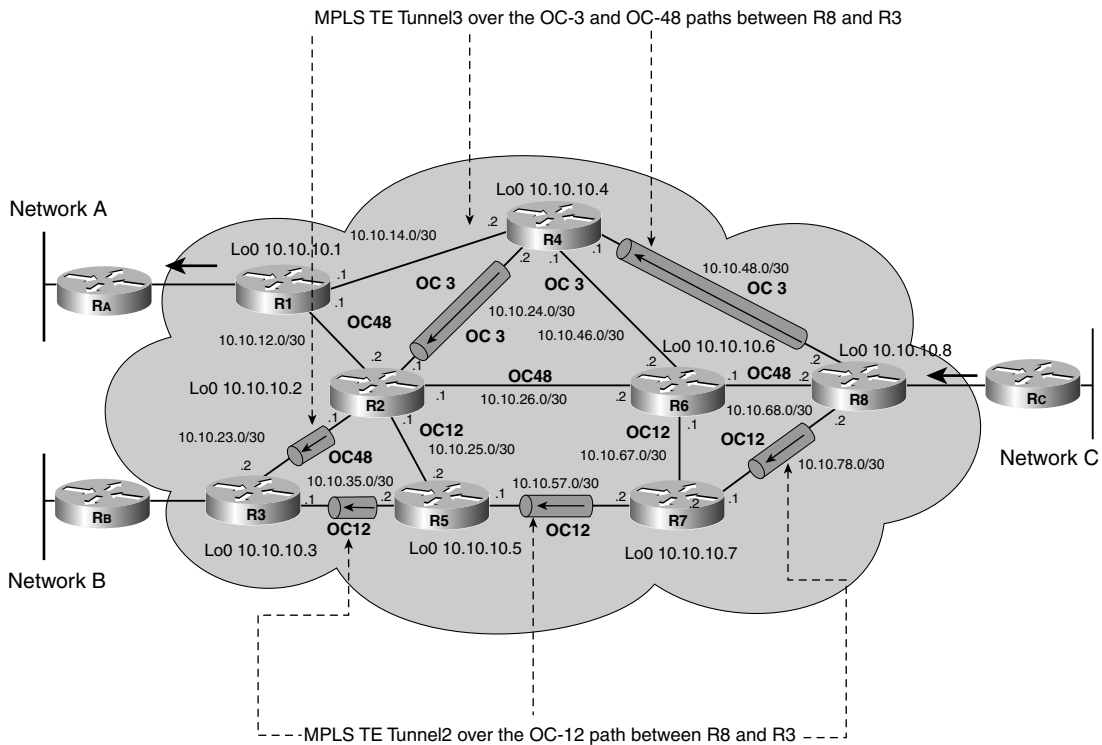


Figure 7-12 R8 to R3 Traffic Engineering Tunnels

Tunnel3 has been configured to utilize R8-R4-R2-R3 (the OC3 path) as the first path (in order of priority) and R8-R7-R5-R3 (the OC12 path) as the second path (in order of priority). The dynamic path is the fallback path if the first and second paths are unavailable due to link or node failure. The dynamic path is normally derived from the IGP, which, in this case, is IS-IS.

The network has also been traffic-engineered to load-balance across Tunnel2 and Tunnel3. The load balancing is achieved by configuring bandwidth statements within each tunnel interface. The ratio of these values is used by CEF to make load-balancing decisions.

R8 Configuration (IS-IS)

The configuration of R8 is as follows:

```
!
hostname R8
!
```

```
ip cef
mpls traffic-eng tunnels
!
interface Loopback0
 ip address 10.10.10.8 255.255.255.255
 ip router isis
!
interface Tunnel0
 ip unnumbered Loopback0
 tunnel destination 10.10.10.1
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 1 1
 tunnel mpls traffic-eng bandwidth 120000
 tunnel mpls traffic-eng path-option 10 explicit name oc12pathR1
 tunnel mpls traffic-eng path-option 20 explicit name oc3pathR1
 tunnel mpls traffic-eng path-option 30 dynamic
!
interface Tunnel1
 ip unnumbered Loopback0
 tunnel destination 10.10.10.1
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 2 2
 tunnel mpls traffic-eng bandwidth 30000
 tunnel mpls traffic-eng path-option 10 explicit name oc3pathR1
 tunnel mpls traffic-eng path-option 20 explicit name oc12pathR1
 tunnel mpls traffic-eng path-option 30 dynamic
!
interface Tunnel2
 ip unnumbered Loopback0
 tunnel destination 10.10.10.3
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng priority 1 1
 tunnel mpls traffic-eng bandwidth 120000
 tunnel mpls traffic-eng path-option 10 explicit name oc12pathR3
 tunnel mpls traffic-eng path-option 20 explicit name oc3pathR3
 tunnel mpls traffic-eng path-option 30 dynamic
!
interface Tunnel3
 ip unnumbered Loopback0
 tunnel destination 10.10.10.3
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 2 2
 tunnel mpls traffic-eng bandwidth 30000
 tunnel mpls traffic-eng path-option 10 explicit name oc3pathR3
 tunnel mpls traffic-eng path-option 20 explicit name oc12pathR3
 tunnel mpls traffic-eng path-option 30 dynamic
!
interface atm4/0/0
 no ip address
```

continues

```
no ip directed broadcast
no atm ilmi-keepalive
!
interface atm4/0/0.1 point-to-point
description OC48 to R6
bandwidth 2500000
ip address 10.10.68.2 255.255.255.252
ip router isis
tag-switching ip
mpls traffic-eng tunnels
pvc 6/9
encapsulation aal5snap
ip rsvp bandwidth 500000 500000
!
interface atm4/0/1
no ip address
no ip directed broadcast
no atm ilmi-keepalive
!
interface atm4/0/1.1 point-to-point
description OC12 to R7
bandwidth 622000
ip address 10.10.78.2 255.255.255.252
ip router isis
tag-switching ip
mpls traffic-eng tunnels
pvc 7/9
encapsulation aal5snap
ip rsvp bandwidth 120000 120000
!
interface atm4/0/2
no ip address
no ip directed broadcast
no atm ilmi-keepalive
!
interface atm4/0/2.1 point-to-point
description OC3 to R4
bandwidth 155000
ip address 10.10.48.2 255.255.255.252
ip router isis
tag-switching ip
mpls traffic-eng tunnels
pvc 8/9
encapsulation aal5snap
ip rsvp bandwidth 30000 30000
!
router isis
net 49.0008.0000.0000.0008.00
is-type level-1
metric-style wide
mpls traffic-eng router-id Loopback0
mpls traffic-eng level-1
```

```
!  
ip classless  
!  
ip explicit-path name oc12pathR1 enable  
next-address 10.10.78.1  
next-address 10.10.57.1  
next-address 10.10.25.1  
next-address 10.10.12.1  
!  
ip explicit-path name oc3pathR1 enable  
next-address 10.10.48.1  
next-address 10.10.14.1  
!  
ip explicit-path name oc12pathR3 enable  
next-address 10.10.78.1  
next-address 10.10.57.1  
next-address 10.10.35.1  
!  
ip explicit-path name oc3pathR3 enable  
next-address 10.10.48.1  
next-address 10.10.24.1  
next-address 10.10.23.1  
!  
end
```

R2 Configuration (IS-IS)

The configuration of R2 is as follows:

```
!  
hostname R2  
!  
ip cef  
mpls traffic-eng tunnels  
!  
interface Loopback0  
ip address 10.10.10.2 255.255.255.255  
ip router isis  
!  
interface atm4/0/0  
no ip address  
no ip directed broadcast  
no atm ilmi-keepalive  
!  
interface atm4/0/0.1 point-to-point  
description OC48 to R1  
bandwidth 2500000  
ip address 10.10.12.2 255.255.255.252  
ip router isis
```

continues

```
tag-switching ip
mpls traffic-eng tunnels
pvc 2/5
encapsulation aal5snap
ip rsvp bandwidth 500000 500000
!
interface atm4/0/1
no ip address
no ip directed broadcast
no atm ilmi-keepalive
!
interface atm4/0/1.1 point-to-point
description OC48 to R3
bandwidth 2500000
ip address 10.10.23.1 255.255.255.252
ip router isis
tag-switching ip
mpls traffic-eng tunnels
pvc 4/6
encapsulation aal5snap
ip rsvp bandwidth 500000 500000
!
interface atm4/0/2
no ip address
no ip directed broadcast
no atm ilmi-keepalive
!
interface atm4/0/2.1 point-to-point
description OC3 to R4
bandwidth 155000
ip address 10.10.24.1 255.255.255.252
ip router isis
tag-switching ip
mpls traffic-eng tunnels
pvc 6/5
encapsulation aal5snap
ip rsvp bandwidth 30000 30000
!
interface atm4/0/3
no ip address
no ip directed broadcast
no atm ilmi-keepalive
!
interface atm4/0/3.1 point-to-point
description OC48 to R6
bandwidth 2500000
ip address 10.10.26.1 255.255.255.252
ip router isis
tag-switching ip
mpls traffic-eng tunnels
pvc 7/9
encapsulation aal5snap
```

```
ip rsvp bandwidth 500000 500000
!
interface atm4/0/4
no ip address
no ip directed broadcast
no atm ilmi-keepalive
!
interface atm4/0/4.1 point-to-point
description OC12 to R5
bandwidth 622000
ip address 10.10.25.1 255.255.255.252
ip router isis
tag-switching ip
mpls traffic-eng tunnels
pvc 8/5
encapsulation aal5snap
ip rsvp bandwidth 120000 120000
!
router isis
net 49.0002.0000.0000.0002.00
is-type level-1
metric-style wide
mpls traffic-eng router-id Loopback0
mpls traffic-eng level-1
!
ip classless
!
end
```

R4 Configuration (IS-IS)

The configuration of R4 is as follows:

```
!
hostname R4
!
ip cef
mpls traffic-eng tunnels
!
interface Loopback0
ip address 10.10.10.4 255.255.255.255
ip router isis
!
interface atm4/0/0
no ip address
no ip directed broadcast
no atm ilmi-keepalive
!
interface atm4/0/0.1 point-to-point
```

continues

```
description OC3 to R1
bandwidth 155000
ip address 10.10.14.2 255.255.255.252
ip router isis
tag-switching ip
mpls traffic-eng tunnels
pvc 3/5
encapsulation aal5snap
ip rsvp bandwidth 30000 30000
!
interface atm4/0/1
no ip address
no ip directed broadcast
no atm ilmi-keepalive
!
interface atm4/0/1.1 point-to-point
description OC3 to R2
bandwidth 155000
ip address 10.10.24.2 255.255.255.252
ip router isis
tag-switching ip
mpls traffic-eng tunnels
pvc 6/5
encapsulation aal5snap
ip rsvp bandwidth 30000 30000
!
interface atm4/0/2
no ip address
no ip directed broadcast
no atm ilmi-keepalive
!
interface atm4/0/2.1 point-to-point
description OC3 to R6
bandwidth 155000
ip address 10.10.46.1 255.255.255.252
ip router isis
tag-switching ip
mpls traffic-eng tunnels
pvc 10/7
encapsulation aal5snap
ip rsvp bandwidth 30000 30000
!
interface atm4/0/3
no ip address
no ip directed broadcast
no atm ilmi-keepalive
!
interface atm4/0/3.1 point-to-point
description OC3 to R8
bandwidth 155000
ip address 10.10.48.1 255.255.255.252
ip router isis
```

```
tag-switching ip
mpls traffic-eng tunnels
pvc 12/3
encapsulation aal5snap
ip rsvp bandwidth 30000 30000
!
router isis
net 49.0004.0000.0000.0004.00
is-type level-1
metric-style wide
mpls traffic-eng router-id Loopback0
mpls traffic-eng level-1
!
ip classless
!
end
```

R5 Configuration (IS-IS)

The configuration of R5 is as follows:

```
!
hostname R5
!
ip cef
mpls traffic-eng tunnels
!
interface Loopback0
ip address 10.10.10.5 255.255.255.255
ip router isis
!
interface atm4/0/0
no ip address
no ip directed broadcast
no atm ilmi-keepalive
!
interface atm4/0/0.1 point-to-point
description OC12 to R3
bandwidth 622000
ip address 10.10.35.2 255.255.255.252
ip router isis
tag-switching ip
mpls traffic-eng tunnels
pvc 5/8
encapsulation aal5snap
ip rsvp bandwidth 120000 120000
!
interface atm4/0/1
```

continues

```
no ip address
no ip directed broadcast
no atm ilmi-keepalive
!
interface atm4/0/1.1 point-to-point
description OC12 to R2
bandwidth 622000
ip address 10.10.25.2 255.255.255.252
ip router isis
tag-switching ip
mpls traffic-eng tunnels
pvc 8/5
encapsulation aal5snap
ip rsvp bandwidth 120000 120000
!
interface atm4/0/2
no ip address
no ip directed broadcast
no atm ilmi-keepalive
!
interface atm4/0/2.1 point-to-point
description OC12 to R7
bandwidth 622000
ip address 10.10.57.1 255.255.255.252
ip router isis
tag-switching ip
mpls traffic-eng tunnels
pvc 15/1
encapsulation aal5snap
ip rsvp bandwidth 120000 120000
!
router isis
net 49.0005.0000.0000.0005.00
is-type level-1
metric-style wide
mpls traffic-eng router-id Loopback0
mpls traffic-eng level-1
!
ip classless
!
end
```

R6 Configuration (IS-IS)

The configuration of R6 is as follows:

```
!
hostname R6
!
ip cef
```

```
mpls traffic-eng tunnels
!
interface Loopback0
 ip address 10.10.10.6 255.255.255.255
 ip router isis
!
interface atm4/0/0
 no ip address
 no ip directed broadcast
 no atm ilmi-keepalive
!
interface atm4/0/0.1 point-to-point
 description OC48 to R2
 bandwidth 2500000
 ip address 10.10.26.2 255.255.255.252
 ip router isis
 tag-switching ip
 mpls traffic-eng tunnels
 pvc 7/9
 encapsulation aal5snap
 ip rsvp bandwidth 500000 500000
!
interface atm4/0/1
 no ip address
 no ip directed broadcast
 no atm ilmi-keepalive
!
interface atm4/0/1.1 point-to-point
 description OC12 to R7
 bandwidth 622000
 ip address 10.10.67.1 255.255.255.252
 ip router isis
 tag-switching ip
 mpls traffic-eng tunnels
 pvc 17/7
 encapsulation aal5snap
 ip rsvp bandwidth 120000 120000
!
interface atm4/0/2
 no ip address
 no ip directed broadcast
 no atm ilmi-keepalive
!
interface atm4/0/2.1 point-to-point
 description OC3 to R4
 bandwidth 155000
 ip address 10.10.46.2 255.255.255.252
 ip router isis
 tag-switching ip
 mpls traffic-eng tunnels
 pvc 10/7
 encapsulation aal5snap
```

continues

```
 ip rsvp bandwidth 30000 30000
 !
 interface atm4/0/3
  no ip address
  no ip directed broadcast
  no atm ilmi-keepalive
 !
 interface atm4/0/3.1 point-to-point
  description OC48 to R8
  bandwidth 2500000
  ip address 10.10.68.1 255.255.255.252
  ip router isis
  tag-switching ip
  mpls traffic-eng tunnels
  pvc 8/9
  encapsulation aal5snap
  ip rsvp bandwidth 500000 500000
 !
 router isis
  net 49.0006.0000.0000.0006.00
  is-type level-1
  metric-style wide
  mpls traffic-eng router-id Loopback0
  mpls traffic-eng level-1
 !
 ip classless
 !
 end
```

R7 Configuration (IS-IS)

The configuration of R7 is as follows:

```
 !
 hostname R7
 !
 ip cef
 mpls traffic-eng tunnels
 !
 interface Loopback0
  ip address 10.10.10.7 255.255.255.255
  ip router isis
 !
 interface atm4/0/0
  no ip address
  no ip directed broadcast
  no atm ilmi-keepalive
 !
 interface atm4/0/0.1 point-to-point
  description OC12 to R5
```

```
bandwidth 622000
ip address 10.10.57.2 255.255.255.252
ip router isis
tag-switching ip
mpls traffic-eng tunnels
pvc 15/1
encapsulation aal5snap
ip rsvp bandwidth 120000 120000
!
interface atm4/0/1
no ip address
no ip directed broadcast
no atm ilmi-keepalive
!
interface atm4/0/1.1 point-to-point
description OC12 to R6
bandwidth 622000
ip address 10.10.67.2 255.255.255.252
ip router isis
tag-switching ip
mpls traffic-eng tunnels
pvc 17/7
encapsulation aal5snap
ip rsvp bandwidth 120000 120000
!
interface atm4/0/2
no ip address
no ip directed broadcast
no atm ilmi-keepalive
!
interface atm4/0/2.1 point-to-point
description OC12 to R8
bandwidth 622000
ip address 10.10.78.1 255.255.255.252
ip router isis
tag-switching ip
mpls traffic-eng tunnels
pvc 11/4
encapsulation aal5snap
ip rsvp bandwidth 120000 120000
!
router isis
net 49.0007.0000.0000.0007.00
is-type level-1
metric-style wide
mpls traffic-eng router-id Loopback0
mpls traffic-eng level-1
!
ip classless
!
end
```

Configuration Case Study of an MPLS Traffic-Engineered Network (OSPF)

In this case study, the same network has been reconfigured to run OSPF as the IGP. The configurations for R1, R3, and R8 are included in this section. The MPLS tunnel headend configurations are similar, except for the OSPF configuration.

NOTE Currently, the Cisco IOS 12.0(s), 12.1, and 12.1T MPLS traffic engineering implementations for OSPF support only single-area OSPF networks.

R1 Configuration (OSPF)

The configuration of R1 is as follows:

```
!
hostname R1
!
ip cef
mpls traffic-eng tunnels
!
interface Loopback0
 ip address 10.10.10.1 255.255.255.255
!
interface Tunnel0
 ip unnumbered Loopback0
 tunnel destination 10.10.10.8
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 1 1
 tunnel mpls traffic-eng bandwidth 120000
 tunnel mpls traffic-eng path-option 10 explicit name r1r8_oc12path
 tunnel mpls traffic-eng path-option 20 explicit name r1r8_oc3path
 tunnel mpls traffic-eng path-option 30 dynamic
!
interface Tunnel1
 ip unnumbered Loopback0
 tunnel destination 10.10.10.8
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 2 2
 tunnel mpls traffic-eng bandwidth 30000
 tunnel mpls traffic-eng path-option 10 explicit name r1r8_oc3path
 tunnel mpls traffic-eng path-option 20 explicit name r1r8_oc12path
 tunnel mpls traffic-eng path-option 30 dynamic
!
interface atm4/0/0
 no ip address
```

```
no ip directed broadcast
no atm ilmi-keepalive
!
interface atm4/0/0.1 point-to-point
description OC48 to R2
bandwidth 2500000
ip address 10.10.12.1 255.255.255.252
tag-switching ip
mpls traffic-eng tunnels
pvc 2/5
encapsulation aal5snap
ip rsvp bandwidth 500000 500000
!
interface atm4/0/1
no ip address
no ip directed broadcast
no atm ilmi-keepalive
!
interface atm4/0/1.1 point-to-point
description OC3 to R4
bandwidth 155000
ip address 10.10.14.1 255.255.255.252
tag-switching ip
mpls traffic-eng tunnels
pvc 3/5
encapsulation aal5snap
ip rsvp bandwidth 30000 30000
!
router ospf 1
network 10.0.0.0 0.255.255.255 area 0
mpls traffic-eng area 0
mpls traffic-eng router-id loop0
!
ip classless
!
ip explicit-path name r1r8_oc12path enable
next-address 10.10.12.2
next-address 10.10.25.2
next-address 10.10.57.2
next-address 10.10.78.2
!
ip explicit-path name r1r8_oc3path enable
next-address 10.10.14.2
next-address 10.10.48.2
!
end
```

R3 Configuration (OSPF)

The configuration of R3 is as follows:

```
!  
hostname R3  
!  
ip cef  
mpls traffic-eng tunnels  
!  
interface Loopback0  
  ip address 10.10.10.3 255.255.255.255  
!  
interface Tunnel0  
  ip unnumbered Loopback0  
  tunnel destination 10.10.10.8  
  tunnel mode mpls traffic-eng  
  tunnel mpls traffic-eng autoroute announce  
  tunnel mpls traffic-eng priority 1 1  
  tunnel mpls traffic-eng bandwidth 120000  
  tunnel mpls traffic-eng path-option 10 explicit name r3r8_oc12path  
  tunnel mpls traffic-eng path-option 20 explicit name r3r8_oc3path  
  tunnel mpls traffic-eng path-option 30 dynamic  
!  
interface Tunnel1  
  ip unnumbered Loopback0  
  tunnel destination 10.10.10.8  
  tunnel mode mpls traffic-eng  
  tunnel mpls traffic-eng autoroute announce  
  tunnel mpls traffic-eng priority 2 2  
  tunnel mpls traffic-eng bandwidth 300000  
  tunnel mpls traffic-eng path-option 10 explicit name r3r8_oc3path  
  tunnel mpls traffic-eng path-option 20 explicit name r3r8_oc12path  
  tunnel mpls traffic-eng path-option 30 dynamic  
!  
interface atm4/0/0  
  no ip address  
  no ip directed broadcast  
  no atm ilmi-keepalive  
!  
interface atm4/0/0.1 point-to-point  
  description OC48 to R2  
  bandwidth 2500000  
  ip address 10.10.23.2 255.255.255.252  
  tag-switching ip  
  mpls traffic-eng tunnels  
  pvc 4/6  
  encapsulation aal5snap  
  ip rsvp bandwidth 500000 500000  
!  
interface atm4/0/1  
  no ip address  
  no ip directed broadcast
```

```
no atm ilmi-keepalive
!
interface atm4/0/1.1 point-to-point
description OC12 to R5
bandwidth 622000
ip address 10.10.35.1 255.255.255.252
tag-switching ip
mpls traffic-eng tunnels
pvc 5/8
encapsulation aal5snap
ip rsvp bandwidth 120000 120000
!
router ospf 1
network 10.0.0.0 0.255.255.255 area 0
mpls traffic-eng area 0
mpls traffic-eng router-id loop0
!
ip classless
!
ip explicit-path name r3r8_oc12path enable
next-address 10.10.35.2
next-address 10.10.57.2
next-address 10.10.78.2
!
ip explicit-path name r3r8_oc3path enable
next-address 10.10.23.1
next-address 10.10.24.2
next-address 10.10.48.2
!
end
```

R8 Configuration (OSPF)

The configuration of R8 is as follows:

```
!
hostname R8
!
ip cef
mpls traffic-eng tunnels
!
interface Loopback0
ip address 10.10.10.8 255.255.255.255
!
interface Tunnel0
ip unnumbered Loopback0
tunnel destination 10.10.10.1
tunnel mode mpls traffic-eng
tunnel mpls traffic-eng autoroute announce
```

continues

```
tunnel mpls traffic-eng priority 1 1
tunnel mpls traffic-eng bandwidth 120000
tunnel mpls traffic-eng path-option 10 explicit name r8r1_oc12path
tunnel mpls traffic-eng path-option 20 explicit name r8r1_oc3path
tunnel mpls traffic-eng path-option 30 dynamic
!
interface Tunnel1
 ip unnumbered Loopback0
 tunnel destination 10.10.10.1
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 2 2
 tunnel mpls traffic-eng bandwidth 30000
 tunnel mpls traffic-eng path-option 10 explicit name r8r1_oc3path
 tunnel mpls traffic-eng path-option 20 explicit name r8r1_oc12path
 tunnel mpls traffic-eng path-option 30 dynamic
!
interface Tunnel2
 ip unnumbered Loopback0
 tunnel destination 10.10.10.3
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng priority 1 1
 tunnel mpls traffic-eng bandwidth 120000
 tunnel mpls traffic-eng path-option 10 explicit name r8r3_oc12path
 tunnel mpls traffic-eng path-option 20 explicit name r8r3_oc3path
 tunnel mpls traffic-eng path-option 30 dynamic
!
interface Tunnel3
 ip unnumbered Loopback0
 tunnel destination 10.10.10.3
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 2 2
 tunnel mpls traffic-eng bandwidth 30000
 tunnel mpls traffic-eng path-option 10 explicit name r8r3_oc3path
 tunnel mpls traffic-eng path-option 20 explicit name r8r3_oc12path
 tunnel mpls traffic-eng path-option 30 dynamic
!
interface atm4/0/0
 no ip address
 no ip directed broadcast
 no atm ilmi-keepalive
!
interface atm4/0/0.1 point-to-point
 description OC48 to R6
 bandwidth 2500000
 ip address 10.10.68.2 255.255.255.252
 tag-switching ip
 mpls traffic-eng tunnels
 pvc 8/9
 encapsulation aal5snap
 ip rsvp bandwidth 500000 500000
```

```
!  
interface atm4/0/1  
  no ip address  
  no ip directed broadcast  
  no atm ilmi-keepalive  
!  
interface atm4/0/1.1 point-to-point  
  description OC12 to R7  
  bandwidth 622000  
  ip address 10.10.78.2 255.255.255.252  
  tag-switching ip  
  mpls traffic-eng tunnels  
  pvc 11/4  
  encapsulation aal5snap  
  ip rsvp bandwidth 120000 120000  
!  
interface atm4/0/2  
  no ip address  
  no ip directed broadcast  
  no atm ilmi-keepalive  
!  
interface atm4/0/2.1 point-to-point  
  description OC3 to R4  
  bandwidth 155000  
  ip address 10.10.48.2 255.255.255.252  
  tag-switching ip  
  mpls traffic-eng tunnels  
  pvc 12/3  
  encapsulation aal5snap  
  ip rsvp bandwidth 30000 30000  
!  
router ospf 1  
  network 10.0.0.0 0.255.255.255 area 0  
  mpls traffic-eng area 0  
  mpls traffic-eng router-id loop0  
!  
ip classless  
!  
ip explicit-path name r8r1_oc12path enable  
  next-address 10.10.78.1  
  next-address 10.10.57.1  
  next-address 10.10.25.1  
  next-address 10.10.12.1  
!  
ip explicit-path name r8r1_oc3path enable  
  next-address 10.10.48.1  
  next-address 10.10.14.1  
!  
ip explicit-path name r8r3_oc12path enable  
  next-address 10.10.78.1  
  next-address 10.10.57.1
```

continues

```
next-address 10.10.35.1
!
ip explicit-path name r8r3_oc3path enable
next-address 10.10.48.1
next-address 10.10.24.1
next-address 10.10.23.1
!
end
```

Summary

For a service provider to truly and successfully implement commercial IP services, a hard QoS with guaranteed delivery of packets is required. This can be accomplished by deploying MPLS traffic engineering across the core backbone. Traffic engineering encompasses many aspects of network performance. These include the provisioning of a guaranteed hard QoS, improving the utilization of network resources by distributing traffic evenly across network links, and providing for quick recovery when a node or link fails.

Unequal-cost load balancing is a concept that allows routers to take advantage of load sharing over multiple unequal-cost paths to a given destination. This can be achieved by manipulating the parameters that determine the routing metrics for protocols such as OSPF, IS-IS, and EIGRP. However, changing a link's metric can potentially change the path of all packets traversing the link. These methods do not provide dynamic redundancy and do not consider the characteristics of offered traffic and network capacity constraints when making routing decisions.

MPLS traffic engineering allows an MPLS backbone to replicate and expand upon the traffic engineering capabilities of Layer 2 ATM and Frame Relay networks. Traffic engineering is essential for service provider and Internet service provider backbones. Both backbones must support a high use of transmission capacity, and the networks must be very resilient so that they can withstand link or node failures.

MPLS traffic engineering allows service providers to define explicit paths, similar to source routing, across their network and steer traffic over these paths. Redundant explicit paths can be configured, thereby providing a fallback mechanism. Furthermore, a final fallback could be configured. This is typically a dynamic path selected by the IGP. Traffic engineering can also perform CEF-based unequal-cost load balancing across tunnels.

MPLS traffic engineering uses RSVP to automatically establish and maintain a tunnel across the backbone. The path used by a given tunnel at any point in time is determined based on the tunnel resource requirements and network resources, such as bandwidth. Available resource information is flooded via extensions to a link-state-based IGP such as OSPF or IS-IS. The integrated routing feature accomplishes automatic assignment of traffic to tunnels using a modified SPF algorithm.

Fast rerouting is a mechanism that minimizes service disruptions for traffic flows affected by an outage, while allowing optimized rerouting servers to reoptimize traffic flows affected by a change in topology. In MPLS, the splicing and stacking techniques are utilized to enable local repair of LSP tunnels.