CHAPTER

ROUTING AND TRAFFIC ENGINEERING USING MPLS

23

If one sticks too rigidly to one's principles, one would hardly see anybody.
Agatha Christie

READING GUIDELINE

Understanding of the material presented in this chapter requires a basic knowledge of MPLS as presented in Chapter 22. In addition, material on IP traffic engineering (Chapter 7), network flow modeling (Chapter 4), and quality-of-service routing (Chapter 21) for the three classes of problems considered is helpful. Note that each problem class should be read independently having the background material from the respective chapters identified here.

Network Routing. DOI: 10.1016/B978-0-12-800737-2.00027-2

In this chapter, we present the applicability of MPLS for routing and traffic engineering for a set of representative real-world problems. Specifically, four problem classes are considered: 1) An integrated IP/MPLS environment for IP traffic engineering; 2) VPN traffic engineering/routing using MPLS; 3) multicast VPN traffic engineering with MPLS; and 4) a Voice over MPLS network, that is, an MPLS network where voice or multimedia real-time interactive service is provided. Our discussion here is primarily limited to intra-domain traffic engineering.

MPLS is usable in a variety of different ways. It is, however, important to understand *how* it is used so that for any future network design or services deployed, it is possible to explore if and how MPLS can be used.

23.1 TRAFFIC ENGINEERING OF IP/MPLS NETWORKS

In this section, we discuss traffic engineering of IP networks, where IP/MPLS integrated routers are deployed. Note that IP traffic engineering is discussed in Chapter 7.

23.1.1 A BRISK WALK BACK IN HISTORY

We will first provide a short historical context for the emergence of MPLS for IP traffic engineering. This is only a very brief overview focusing on a few key facts in regard to IP traffic engineering and is not focused on providing precise details of what happened when. For a detailed history of MPLS and its forerunners, refer to [211], [321].

By the mid to late 1990s, it was realized that some form of traffic engineering of IP networks was needed; at that time, large IP networks were either using OSPF or IS–IS protocol and, primarily, either simple hop-based link weight or the inverse of link capacity as link weights. As we discussed earlier in Chapter 7, there are many network situations where it is possible to have some links with very high utilization if the link weights are not assigned properly.

Somewhat independent of the above development, there were concerns about the IP forwarding engine's ability to handle a large volume of traffic. Concepts such as IP switching, tag switching, and aggregate router-based IP switching (ARIS) emerged in 1996. It was soon recognized that a standard switching approach for packets was needed, which led to the MPLS workgroup being chartered by IETF in early 1997. By 1999, the role of MPLS in IP traffic engineering was well recognized [60], [63], citing the limitation of OSPF/IS-IS in being able to move traffic away from heavily utilized links due to lack of any control mechanism.

By 2000, however, it was reported that there was indeed a systematic way to determine OSPF/IS-IS link weights for IP traffic engineering [284], [667] (see also [285], [668]). Certainly, this was good news for many ISPs who wanted to continue to run IP-only routers in their network. Six years later, many large ISPs continue to successfully run IP-only networks with good traffic engineering through optimal link weight determination coupled with good traffic matrix determination.

Certainly, MPLS has its place in IP traffic engineering; in fact, many other large ISPs successfully run IP/MPLS networks for controlled IP traffic engineering. It is also important to note that MPLS has now found roles in many arenas, as discussed in Chapter 22. Thus, whether IP-only is better or worse than IP/MPLS for IP traffic engineering is often a matter of opinion and preference; furthermore, this

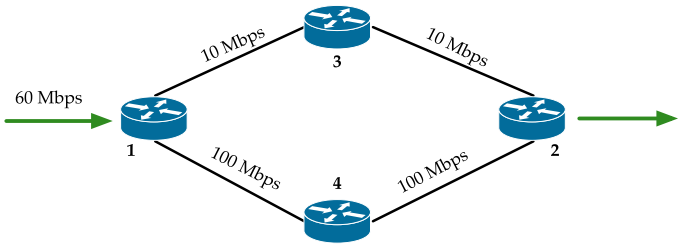


FIGURE 23.1

4-node example with different link capacity.

is also tied to customers that a provider is serving as well personnel and expertise locally available. We will next present the essence of IP/MPLS traffic engineering in a provider's network in its own right.

23.1.2 MPLS-BASED APPROACH FOR TRAFFIC ENGINEERING

The basic question is how to control traffic movement through a network if we do not like current traffic flows on different links. Ideally, it is desirable to somehow force traffic to a certain path. This is where one of the benefits of MPLS comes into the picture; that is, an LSP can be set up, where desired and when desired, and the bandwidth flow can be limited.

First note that once a tunnel is set up through MPLS, it appears as a logical link at the IP level, especially to the routing protocol. In an IP/MPLS network, the routing protocols such as OSPF and IS-IS are used in extension mode, i.e., OSPF-TE or IS-IS-TE, which provides bandwidth information. This information is then used by the traffic engineering component to determine LSPs.

When Traffic Demand Is Fixed

We first start with an example in which the traffic demand is fixed and given.

Example 23.1. A simple example of IP/MPLS traffic engineering.

We first discuss the 4-node example presented in Chapter 7 for IP traffic engineering; the topology along with demand and link capacity is reproduced in Figure 23.1. We assume that the goal is to minimize maximum link utilization; then, the optimal solution is to send $100/(100+10)$ ratio of traffic volume, i.e., 54.55 Mbps of the total east-west traffic on the south route and the rest of the 5.45 Mbps on the north route.

In the case with IP traffic engineering, the best we can do is to allow all flows, 60 Mbps, to take the south route, which is accomplished by choosing link weights so that the cost of the south route is lower than the north route. Now consider employing MPLS on this same network. We can now set up two LSPs, one with the guaranteed bandwidth set to 54.55 Mbps (south path) and the other to 5.45 Mbps (north path), while the MPLS router on the left provides proportional load balancing in terms of packet flow. ●

On Determining Optimal LSPs

To determine where and how many endpoint tunnels (from ingress to egress) are needed, the multicommodity network flow (MCNF) model presented in Eq. (4.4.10) can be used; note the subtle difference with the multicommodity shortest-path routing flow (MCSPRF) model Eq. (7.5.1). Recall that Eq. (4.4.10) is shown for minimizing maximum link utilization; as appropriate, the objective function can be replaced by either a piece-wise linear approximation of the delay function or by a composite function; refer to Eq. (7.6.29) and Eq. (7.6.24), respectively. Once the MCNF model is solved, we will obtain a solution where paths with positive flows will be identified; these paths are then prime candidates for becoming LSP tunnels in the IP/MPLS network. Note that the MCNF model may identify some paths with very small positive flow amounts; such paths may not need to be considered in the final optimal LSPs selected.

As an alternative to the MCNF approach, a constrained shortest-path first approach for LSP determination can be taken. Typically, this approach cannot provide optimal flows under certain situations. This will be illustrated later in the context of MPLS VPN traffic engineering.

Time-Dependent Traffic Variation

Traffic does change, for example, in a 24-hour time cycle quite significantly. Instead of considering a single traffic matrix, multiple traffic matrices for different hours during the day may be estimated. Thus, the MCNF model can be run on each of these matrices separately. The resulting paths with positive flows for each such traffic matrix is very likely going to be different. Yet, some paths will be common. If so, such paths can be candidates for LSP tunnels to be set up as explicit LSP routes where the bandwidth allocation can be varied from one time period to another. For the ones not common, LSPs can be set up on a time-dependent basis. An important issue to keep in mind is that tearing down and setting up LSPs can affect the end-user's performance. Thus, minimizing such an impact is also important.

When Traffic Demand Is Not Fixed

Next we consider the case in which the traffic demand is not fixed; this case is not to be confused with the case of time-dependent variations in traffic demand. In an IP network, demand is stochastic as it can vary instantaneously; this is sometimes referred to as elastic demand. Thus, from measurements, we can at best determine *projected* demand, not fixed demand. Going back to Example 23.1, consider the case where 60 Mbps is the projected demand. The traffic may fluctuate from this value at any instant, possibly going over 60 Mbps. If the network is set up with a guaranteed bandwidth on each LSP, then any traffic over 60 Mbps will be denied entry to the network. This is certainly not a good situation in an IP traffic environment. Therefore, LSPs would need to be set up carefully to allow for fluctuations and also to know that one path has much less bandwidth. For example, once the RSVP-TE Path message is initiated, the LSP on the north path is set up with int-serv under a guaranteed service option to limit it from having to handle any load fluctuations. The LSP on the south path can be set up using RSVP-TE with int-serv under a controlled load service option to allow for flexibility for any overload. Note that there is no instantaneous service or delay guarantee; however, in a best-effort IP network or in a differentiated services IP network, this allows for service level agreements to be met.

An alternate solution is to set up two LSPs on the south path, where one is set up at 54.55 Mbps with a guaranteed option, and the other one is set up with a null service option. Depending on the

implementation, another option is to allow any traffic over 54.55 Mbps to be routed as IP traffic based on the shortest-path first routing decision made by the interior gateway protocols (IGPs). For this to happen, we need to ensure that the links on the north route are set with high weights, so that the IGP does not select this as a preferred route for any overflow traffic. Another point to note is that the ingress node must be able to handle overflow of traffic from the first LSP to the second LSP. That is, at any time in IP/MPLS networks, both link weight setting and MPLS LSP set up is possible; while this provides flexibility, it also results in a certain amount of complexity in determining and managing link weights as well as MPLS LSPs so that the network is efficiently used—this is not an easy problem.

Joint Link Weights and MPLS LSP Engineering

In general, the joint traffic engineering optimization problem determining link weights for the IGP *and* optimal MPLS LSPs is a complex problem for an integrated IP/MPLS network. We will consider a special case here that allows us to approach this problem through decoupling.

Suppose that a large ISP has a set of critical customers with web servers running directly off the large LSP. In this case, the IP address block of the customer's web servers would be known. Because of service level agreements, it is decided that these customers would get specialized treatment. Thus, at the entry points to the network, traffic trunks for such customers can be defined that point to the address block of web servers; accordingly, LSP tunnels can be set up. Thus, when a real user's packet arrives at the ingress router, it will go on a "fast track" LSP to the destination, since such LSP tunnels have already been established for user traffic delivery. Alternately, the maximum rate that is allowed to be handled can also be limited using the same idea. In other words, controlled traffic engineering can be helpful in providing special treatment to large customers.

However, all the rest of the users can use standard IP-mode service. Given this, we can approach the joint optimization somewhat differently through a two-stage approach:

- For customers with SLAs, estimate traffic demands and determines the optimal LSPs using the MCNF approach (refer to Section 4.4.2).
- Determine residual link capacities after allocating bandwidth resources for the required LSP.
- For traffic estimated (that does not fall under SLAs provided through MPLS LSPs), consider the link weight optimization approach using an MCSPRF approach (refer to Section 7.5), in which case link capacity is considered to be the residual link capacity.

In addition, some customers might require failure protection as part of the SLA, which can be supported through the FAST-REROUTE option in MPLS and by providing backup LSPs. In general, customers with varied levels of protection requirements might need to be accommodated through MPLS tunnels. To traffic engineer a network for this requirement, a transport modeling approach can be used; refer to Section 24.5 and see also [783], [875].

Tunnel in the Middle

Finally, it may be noted that through label stacking, MPLS allows LSP tunnels to be set up in the "middle"; see Figure 22.5 for an example of tunnel in the middle. Based on traffic profile and knowledge of a specific network, it is quite possible to consider the option of creating tunnels in the middle.

However, the general problem of selecting tunnels at the end and also determining *where* in the middle is a difficult combinatorial optimization problem.

General Remark

We can see from the above discussion that in an IP/MPLS environment, the traffic engineering approach and decision depend on what types of customers a provider is serving, and the level of guarantee needed for meeting demand volume request.

23.2 VPN TRAFFIC ENGINEERING

It may be noted that VPN is a widely used terminology for a broad variety of VPN services including accessing a corporate network from home through a VPN service. Here, the meaning of VPN is different in that a corporate customer may have offices in different physical locations distributed geographically; such customers would like to lease a seamless virtual link connectivity through another provider ("VPN provider") that has geographic presence in these areas. In this section, we present an MPLS VPN traffic engineering approach for such virtual services. Note that this usage of MPLS is quite different from the IP/MPLS traffic engineering issue for an ISP. Here, our focus is networks provided by MPLS/VPN providers, which are not to be confused with public ISPs. Such a VPN is also known as a provider-provisioned VPN (PPVPN); generic requirements and terminology for PPVPN can be found in [34], [610].

In Chapter 22, we discussed how a provider-provisioned VPN can be accomplished using BGP MPLS. We briefly review a few key points for the purpose of VPN traffic engineering. Here, we consider the case where the connectivity is provided at layer 3, i.e., a layer 3 VPN service. For illustration, we will assume that each customer has its own address block. From the point of view of the VPN provider, it will be necessary to have a label-edge router where a customer is connected at layer 3, and then another LER at the other geographic location to connect back to the customer. Thus, within the VPN provider's networks, the LERs serve as ingress and egress points and the provider can have multiple LSRs for transiting traffic; such VPN providers are referred to as MPLS VPN providers, or more generally, as provider provisioned VPN providers. Note that the ingress and egress points serve as locations for LSP tunnels to originate and terminate to serve different customers. Furthermore, LERs are referred to as *provider edge (PE)* routers while the routers at customer sites are referred to as *customer edge (CE)* routers.

Later, we will consider another VPN concept called *layer 2 VPN* (see Section 22.5.2). In this case, the customer edge is *not* a router. This is discussed later in Section 23.2.4.

23.2.1 PROBLEM ILLUSTRATION: LAYER 3 VPN

We will illustrate a routing/traffic engineering problem from the perspective of an MPLS VPN provider who will be referred to as *ProviderStealth*. This provider has three customers: Customer A, Customer B, and Customer C. Customer A has locations in three cities: San Francisco (SF), Kansas City (KC), and New York (NY), while Customer B and Customer C have locations only in San Francisco and New York. We assume that each of these customers has already obtained an IP address block as follows:

Table 23.1 Customer demand matrix.

Customer ID	Locations between		Bandwidth Requirement
Customer A (27.27.0.0/16)	Kansas City (27.27.1.0/24)	San Francisco (27.27.128.0/24)	45 Mbps
	Kansas City (27.27.1.0/24)	New York (27.27.192.0/24)	60 Mbps
	San Francisco (27.27.128.0/24)	New York (27.27.192.0/24)	20 Mbps
Customer B (42.84.0.0/16)	San Francisco (42.84.0.0/20)	New York (42.84.128.0/20)	80 Mbps
Customer C (2.4.0.0/16)	San Francisco (2.4.0.0/20)	New York (2.4.128.0/20)	100 Mbps

Customer A: 27.27.0.0/16
Customer B: 42.84.0.0/16
Customer C: 2.4.0.0/16

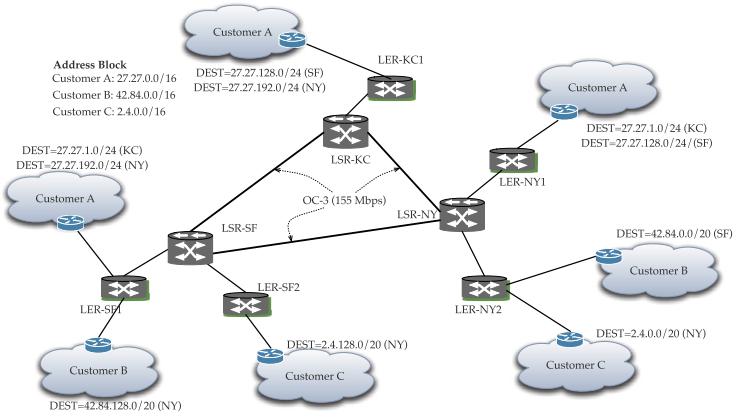
Customer A decides to activate only three subnets at a /24 level: 27.27.1.0/24 for KC, 27.27.128.0/24 for SF, and 27.27.192.0/24 for NY. Customer B has decided to equally divide its address space in its two locations using /20 and, thus, has allocated 42.84.0.0/20 to SF and 42.84.128.0/20 to NY. Customer C has also used the same address allocation rule for its address block, i.e., 2.4.0.0/20 to SF and 2.4.128.0/20 to NY. Each customer has a bandwidth requirement between its different sites as located in Table 23.1.

ProviderStealth has LERs and LSRs at a PoP in each city and customers would need to have connectivity to each PoP's LERs at respective locations; ProviderStealth's responsibility then is to meet the demand requirement of each customer in its MPLS VPN network. ProviderStealth's core network links are assumed to be OC-3 (155 Mbps), which provides an OC-3 rate in each direction. The entire network topological view is shown in Figure 23.2(a). From the bandwidth requirement, we can see the total bandwidth requirement between SF and NY is 200 Mbps; since ProviderStealth has only an OC-3 capacity between SF and NY, it cannot meet the total bandwidth requirement using this direct link. By inspecting its capacity in the entire network, it can route Customer B's requirement through KC taking the path SF to KC to NY using LSRs in each city. Accordingly, ProviderStealth will set up label switched paths for traffic engineering tunneling for Customer B.

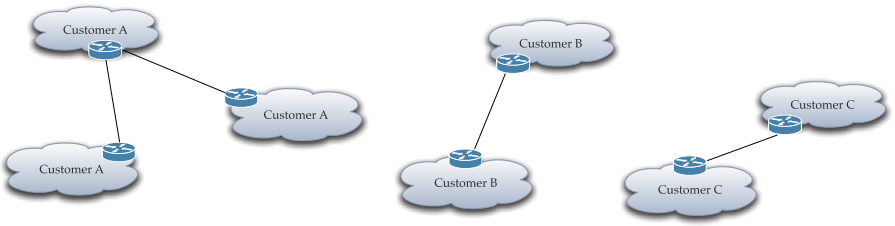
The LSPs in each direction are listed in Table 23.2 where FECs can be assigned based on the network destination for each customer. Note that LSPs are unidirectional; thus, two LSPs must be set up to meet the bidirectional requirement on bandwidth. The routes for the LSPs are shown in Figure 23.2(c), while the logical connectivity view to each customer would be made apparent of the MPLS VPN network by the MPLS VPN provider and is shown for each customer in Figure 23.2(b).

Example 23.2. Customers' private addressing and MPLS VPN.

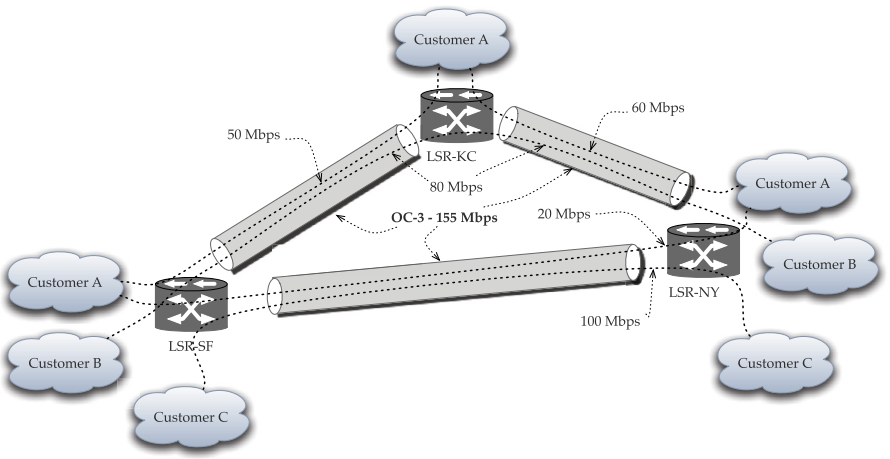
In the above illustration, we used different IP address blocks for different customers. It is now common for organizations to use private IP address blocks such as 10.0.0.0/8 for numbering within their organizations, with different subnets defined for different locations. Because of this, it is possible that two different customers have the same private address subnets, say 10.5.3.0/24 assigned for their own locations. This may look conflicting from the point of view of proper routing within the MPLS



(a) Network connectivity view, including addressing



(b) Logical view to customers



(c) Routing view to MPLS-VPN provider

FIGURE 23.2

MPLS-VPN routing/traffic engineering example.

Table 23.2 LSPs chosen as traffic engineering tunnels.

Customer ID	Origin-Destination	LSP for TE Tunnel
Customer A	SF–KC (for 27.27.1.0/24)	LER-SF1 ··· LSR-SF ··· LSR-KC ··· LER-KC1
	KC–SF (for 27.27.128.0/24)	LER-KC1 ··· LSR-KC ··· LSR-SF ··· LER-SF1
	KC–NY (for 27.27.192.0/24)	LER-KC1 ··· LSR-KC ··· LSR-NY ··· LER-NY1
	NY–KC (for 27.27.1.0/24)	LER-NY1 ··· LSR-NY ··· LSR-KC ··· LER-KC1
	SF–NY (for 27.27.192.0/24)	LER-SF1 ··· LSR-SF ··· LSR-NY ··· LER-NY1
	NY–SF (for 27.27.128.0/24)	LER-NY1 ··· LSR-NY ··· LSR-SF ··· LER-SF1
Customer B	SF–NY (for 42.84.128.0/20)	LER-SF1 ··· LSR-SF ··· LSR-KC ··· LSR-NY ··· LER-NY2
	SF- (for 42.84.0.0/20)	LER-NY2 ··· LSR-NY ··· LSR-KC ··· LSR-SF ··· LER-SF1
Customer C	SF–NY (for 2.4.128.0/20)	LER-SF2 ··· LSR-SF ··· LSR-NY ··· LER-NY2
	SF- (for 2.4.0.0/20)	LER-NY2 ··· LSR-NY ··· LSR-SF ··· LER-SF2

network. However, this is not an issue if BGP/MPLS IP VPN functionality [712], [713], presented in Section 22.5.1, is used, which uses route distinguishers to distinguish between two customer's subnet addresses. Regardless of the numbering issue, the traffic engineering problem faced by the VPN provider is the same as if the address blocks were unique. ♦

23.2.2 LSP PATH DETERMINATION: CONSTRAINED SHORTEST PATH APPROACH

Assume that MPLS routers are equipped with a constrained shortest-path first (CSPF) algorithm that is similar to shortest path algorithm, Algorithm 2.4, described in Chapter 2. There are two main differences/requirements: (1) a link is considered only if it has the bandwidth available to meet the request, and (2) a path must be computed only for a given destination, say, v. A simple way to address the first difference is to prune links that do not meet the bandwidth requirement by temporarily setting the link cost to infinity. For the second requirement, the algorithm needs to stop as soon as the path is found. For completeness, the basic idea of a constrained shortest-path first algorithm is listed in Algorithm 23.1 using the same notation as used in Chapter 2; note that this algorithm is particularly stated for meeting *bandwidth* constraint. Other resource constraints can be considered as well by appropriately changing Step-2 of this algorithm.

To use CSPF for the problem illustrated, we first note that in our case, the link cost may be set to the hop count. The bandwidth availability can be determined at each router based on OSPF-TE or IS-IS-TE for traffic engineering. With this information, a sequence of steps would need to be performed that can be invoked at each router *independently* as follows:

1. Set up TE 100 Mbps tunnel for Customer C (at LSR-SF from SF to NY, and reverse)
   Available link bandwidth: SF–NY: 55 Mbps; SF–KC: 155 Mbps; KC–NY: 155 Mbps
2. Set up TE 80 Mbps tunnel for Customer B (at LSR-SF from SF to NY, and reverse)
   Available link bandwidth: SF–NY: 55 Mbps; SF–KC: 75 Mbps; KC–NY: 75 Mbps
3. Set up TE 20 Mbps tunnel for Customer A (at LSR-SF from SF to NY, and reverse)
   Available link bandwidth: SF–NY: 35 Mbps; SF–KC: 75 Mbps; KC–NY: 75 Mbps
4. Set up TE 45 Mbps tunnel for Customer A (at LSR-SF from SF to KC, and reverse)
   Available link bandwidth: SF–NY: 35 Mbps; SF–KC: 30 Mbps; KC–NY: 75 Mbps
5. Set up TE 60 Mbps tunnel for Customer A (at LSR-KC from KC to NY, and reverse)

Algorithm 23.1 Constrained shortest-path first algorithm: from node i to node v, for bandwidth constraint, computed at time t.

1. Network $\mathcal{N}$ and cost of link $d^i_{km}(t)$ and available bandwidth on $b^i_{km}(t)$ on link k–m, as known to node i at the time of computation, t.
2. For link k–m, if available bandwidth, $b^k_{km}(t)$, is smaller than bandwidth request $\bar{b}$, then set link cost temporarily to infinity, i.e., $d^i_{km}(t) = \infty$.
3. Initially, consider only source node i in the list of nodes considered ("permanent list"), i.e., $\mathcal{S} = \{i\}$; mark the list with all the rest of the nodes as $\mathcal{S}'$ ("tentative list"). Initialize
   $\underline{D}_{ij}(t) = d^i_{ij}(t)$, for all $j \in \mathcal{S}'$.
4. Identify a neighboring node (intermediary) k not in the current list $\mathcal{S}$ with the minimum-cost path from node i, i.e., find $k \in \mathcal{S}'$ such that $\underline{D}_{ik}(t) = \min_{m \in \mathcal{S}'} \underline{D}_{im}(t)$
   if k is the same as destination v, *stop*.
   Add k to permanent list $\mathcal{S}$, i.e., $\mathcal{S} = \mathcal{S} \cup \{k\}$,
   Drop k from tentative list $\mathcal{S}'$, i.e., $\mathcal{S}' = \mathcal{S}' \backslash \{k\}$.
   If $\mathcal{S}'$ is empty, *stop*.
5. Consider neighboring nodes $\mathcal{N}_k$ of the intermediary k (but do not consider nodes already in permanent list $\mathcal{S}$) to check for improvement in the minimum-cost path, i.e.,
   for $j \in \mathcal{N}_k \cap \mathcal{S}'$

$$\underline{D}_{ij}(t) = \min\{\underline{D}_{ij}(t), \underline{D}_{ik}(t) + d^i_{kj}(t)\} \tag{23.2.1}$$

   go to Step-4.

Available link bandwidth: SF–NY: 35 Mbps; SF–KC: 30 Mbps; KC–NY: 15 Mbps

Since MPLS tunnel set-up is undirectional, each direction must be set up separately. The change in *available capacity* at each link after each step is also noted above. Step-2 above requires further explanation. Since after Step-1, link SF–NY has only 55 Mbps left, CSPF will prune this link since it cannot meet the 80 Mbps requirement, which will result is choosing path SF–KC–NY.

Note again that CSPF is performed by each router indendently based on its current view of bandwidth availability. Suppose that requests were submitted and invoked in the following order in which the first two first two steps from the above are swapped:

1′. Set up TE 80 Mbps tunnel for Customer B (at LSR-SF from SF to NY, and reverse)
Available link bandwidth: SF–NY: 75 Mbps; SF–KC: 155 Mbps; KC–NY: 155 Mbps

2′. Set up TE 100 Mbps tunnel for Customer C (at LSR-SF from SF to NY, and reverse)
Available link bandwidth: SF–NY: 75 Mbps; SF–KC: 55 Mbps; KC–NY: 55 Mbps

3. Set up TE 20 Mbps tunnel for Customer A (at LSR-SF from SF to NY, and reverse)
Available link bandwidth: SF–NY: 55 Mbps; SF–KC: 55 Mbps; KC–NY: 55 Mbps

4. Set up TE 45 Mbps tunnel for Customer A (at LSR-SF from SF to KC, and reverse)
Available link bandwidth: SF–NY: 55 Mbps; SF–KC: 10 Mbps; KC–NY: 55 Mbps

Now we can see that after the fourth step, there is not enough unsplit tunnel bandwidth left in the network to accommodate the final request of 60 Mbps. It may be noted that in the above case, you can go back and release the first LSP that was already set up in order to rearrange and fit them all. That is, in most cases, the CSPF approach works quite well; however, the order can matter and it is important to be careful. Otherwise, extra work/steps would be needed to reset some LSP tunnels. This is an issue, in particular, if network bandwidth is tight. If there is plenty of bandwidth, CSPF should

not have trouble finding feasible paths. However, rearrangement can be time consuming to do for a large network, especially if it were to be done at the command line.

23.2.3 LSP PATH DETERMINATION: NETWORK FLOW MODELING APPROACH

In this section, we discuss how to arrive at an optimal traffic engineering solution from the point of view of the MPLS VPN provider using a network flow optimization approach. For the small network example we have discussed, we can use functionalities such as OSPF-TE or IS-IS-TE to obtain bandwidth information about different links, and then issue a tunnel set-up command at the router's command line interface that invokes the constrained shortest-path approach. While this is a doable approach, it is not a scalable approach as the network size grows; in addition, the impact of the order of the CSPF invocation is difficult to predict in a large network.

Thus, in a large network environment, it would be necessary to do global optimization for the best traffic engineering solution. Here, for ease of illustration, we will still consider the same example as in the previous section and discuss how optimization is performed. In addition, the following discussion shows how network flow modeling presented in Chapter 4 can be used for VPN traffic engineering.

The network has a total of eight LERs/LSRs in the ProviderStealth's network, out of which five are LERs. Thus, a simple way to look at it is that we need to consider a 5×5 traffic demand matrix. However, this is often not necessary since instead of using an LER-level view, we can consider a PoP level view. That is, there are three PoPs, one each in San Francisco, Kansas City, and New York. Thus, the core network routing is the key problem here rather than how an LER is connected to an LSR at a particular PoP. Second, the core network links are usually where the capacity is more constrained; here, we have used an OC-3 link. A link between an LER and an LSR in the same PoP may be on a Gigabit Ethernet LAN—certainly, this bandwidth is not as tight of a constraint as the core network link. Thus, we can abstract the problem at the PoP-to-PoP level as a 3-"node" network where there are two paths: direct or via the third PoP.

Thus, we will consider the PoP-to-PoP network problem. We have three distinct customers that we need to track separately. However, we do not need to consider each direct path separately; for the model, bidirectionality can be used that reduces the number of constraints to be considered. After the solution is obtained, the LSPs can be generated based on direction.

To see how to model the problem, consider Customer B for which we need to choose from two possible *candidate* paths: either direct on SF to NY or the alternate one from SF to KC to NY. We can assign two unknowns, Xs, for these two possible paths, and impose the binary requirement that only one of them must be chosen, i.e., the following decision requirement:

```
x_B_sf_ny + x_B_sf_kc_ny = 1
```

where `x_B_sf_ny` can take either the value 0 or 1; this is similar for `x_B_sf_kc_ny`. Certainly, both cannot be 1 in the final solution since that will then violate the above equation. In the same way, we can write for other demands. Since there are five demands (three for Customer A and one each for Customers B and C), we will have a total of five such equations. Note that if we were to consider each direction separately, we would have 10 equations—an unnecessary increase in the number of equations, which becomes very prominent in solving a large network problem. Next, we need to consider the bandwidth constraint on each core link. Consider the OC-3 link with a capacity of 155 Mbps between

SF and KC. This link can be used by any of the paths for each of the customers, as long as the capacity is not exceeded. We need to consider the fact that if a path for a Customer between two locations is chosen, then this path must be allocated the demand requirement. We will use the demand requirement as stated earlier in Table 23.1. If, for example, path `x_A_sf_kc_ny` is chosen, then on each link, SF–KC and KC–NY, 80 Mbps would need to be allocated. Since the unknowns are defined as binary variables we can multiply such a variable by the demand amount. If we now consider all of the possible candidate paths for different customers and locations, we see that for the SF–KC link the following condition must be satisfied:

```
  45 x_A_sf_kc + 60 x_A_kc_sf_ny +  20 x_A_sf_kc_ny
               + 80 x_B_sf_kc_ny + 100 x_C_sf_kc_ny <= 155
```

Since not all Xs can take a value of 1, these capacity constraints have to work in concert with the decision requirements. Finally, an objective function may be considered that is appropriate for the provider. For simplicity, we will assume here that the "cost" of each possible path is one. We can write the entire optimization problem as follows:

```
Minimize x_A_sf_kc + x_A_sf_ny_kc + x_A_kc_ny + x_A_kc_sf_ny
       + x_A_sf_ny + x_A_sf_kc_ny + x_B_sf_ny + x_B_sf_kc_ny
       + x_C_sf_ny + x_C_sf_kc_ny
subject to
 d45_A_sf_kc:   x_A_sf_kc + x_A_sf_ny_kc = 1
 d60_A_kc_ny:   x_A_kc_ny + x_A_kc_sf_ny = 1
 d20_A_sf_ny:   x_A_sf_ny + x_A_sf_kc_ny = 1
 d80_B_sf_ny:   x_B_sf_ny + x_B_sf_kc_ny = 1
 d100_C_sf_ny:  x_C_sf_ny + x_C_sf_kc_ny = 1
 l_sf_kc: 45 x_A_sf_kc +   60 x_A_kc_sf_ny +  20 x_A_sf_kc_ny
    + 80 x_B_sf_kc_ny + 100 x_C_sf_kc_ny <= 155
 l_sf_ny: 45 x_A_sf_ny_kc + 60 x_A_kc_sf_ny + 20 x_A_sf_ny
    +  80 x_B_sf_ny + 100 x_C_sf_ny <= 155
 l_kc_ny: 45 x_A_sf_ny_kc + 60 x_A_kc_ny +   20 x_A_sf_kc_ny
    +  80 x_B_sf_kc_ny + 100 x_C_sf_kc_ny <= 155
Integer
 x_A_sf_kc   x_A_sf_ny_kc   x_A_kc_ny   x_A_kc_sf_ny
 x_A_sf_ny   x_A_sf_kc_ny   x_B_sf_ny   x_B_sf_kc_ny
 x_C_sf_ny   x_C_sf_kc_ny
End
```

Note that the above is the format accepted by CPLEX, a linear optimization tool mentioned in Chapter 4. Note that each decision equation or constraint is identified at the beginning of the line with a name; for ease of tracking, we have embedded the demand value and location/customer information in such names for decision requirements, for example, `d80_B_sf_ny`, and link names, for example, `l_sf_kc`. Also, note that to indicate the binary nature of the path choice, the unknowns must be declared as "Integer," which means binary by default in CPLEX. On solving this model, we obtain the following solution:

```
x_A_sf_kc = 1, x_A_kc_ny = 1, x_A_sf_ny = 1,
x_B_sf_kc_ny = 1, x_C_sf_ny = 1.
```

All the rest of the decision variables are zero. We can see that for Customer B, path SF–KC–NY is selected. Accordingly, this solution can be implemented by generating LSPs in each direction by taking into account the LER-LSR path; this is shown earlier in Table 23.2. It may be noted that this problem does not have a unique solution. For instance, Customer C could have routed on SF–KC–NY instead of Customer B; the capacity constraints will still be satisfied and the objective cost as defined here would be the same. Thus, sometimes additional factors need to be taken into account in defining the objective function such as whether any cost weight should be given to any customer, or on link utilization, or if twice the weight should be placed on two-link paths. Accordingly, the objective function can be adjusted in the above model. For example, if we were to give twice the weight to longer paths, then the objective function will take the following form:

```
Minimize x_A_sf_kc + 2 x_A_sf_ny_kc + x_A_kc_ny + 2 x_A_kc_sf_ny
       + x_A_sf_ny + 2 x_A_sf_kc_ny + x_B_sf_ny + 2 x_B_sf_kc_ny
       + x_C_sf_ny + 2 x_C_sf_kc_ny
```

Note that for the above problem considered, the optimal solution would not change by using this modified objective. We have listed the above objective to illustrate another point. Suppose the fact that unknowns are to be binary is not declared, i.e., the part with "Integer" is left out. What does this mean? This means that decision equations must be satisfied, but each can take fractional values at the solution! In fact, for this problem by ignoring the binary requirement, with the modified objective, we find that the solution for Customer A remains the same. Customer B will be routed on the direct SF–NY route while Customer C's requirement will be split over two paths: 55% on the direct SF–NY path and 45% on the SF–KC–NY path; that is, create a 55-Mbps tunnel on the SF–NY path and another 45-Mbps tunnel on the SF–KC–NY path. Recall our discussion earlier about a traffic trunk being split on two LSPs; this is then an example of how this can be generated through a network flow modeling approach; here, the customer demand requirement is a traffic trunk.

There are several additional points to note:

- For the same objective function considered, the total bandwidth required to accommodate all demands with non-split LSPs is more than with split LSPs. This is an important observation that is a result of linear programming theory: the *integer* linear programming (ILP) solution cost is either equal to or more than the linear programming (LP) solution cost when the same objective function is minimized where the cost coefficient in the objective function is non-negative. For instance, in the above example, with the modified cost function, the split solution results in a total network bandwidth requirement of 350 Mbps as opposed to 385 Mbps for the non-split solution, out of a total bandwidth of $3 \times 155 = 465$ Mbps in the core network.
- The above observation can be used in deciding to split traffic trunks into multiple paths if the network capacity is tight, or to delay capacity expansion cost temporarily.
- The decision to split a traffic trunk for a customer into multiple paths could itself depend on the terms of the SLA with the customer. Accordingly, this requirement can be taken into account in the modeling phase. In particular, for the customers for which a traffic trunk split is allowed, the path variables can be defined using continuous variables, and for the customers where the traffic trunk split is not, the path variables are defined using binary variables, as presented earlier; the network flow modeling framework can handle this mixed-mode scenario, which results in a mixed integer linear programming problem.

- In addition to customer traffic, a network carries control traffic and management traffic. Thus, on each link, a certain amount of bandwidth can be set aside. This can also be incorporated in the network modeling approach. For example, if 10 Mbps is to be set aside on each link for control and management traffic, then the link capacity constraint requirement "<= 155" can be replaced by "<= 145." The same idea can also be used if no links are to be allocated to its fullest capacity in anticipation of future requests.
- The bandwidth requested by a customer may vary depending on time. This scenario occurs when customers have plants in different countries around the globe. When coupled with pricing for such service, a time varying bandwidth requirement may be requested. If so, it would be necessary to do a network reoptimization periodically because of the time-dependent demands; the model discussed above can still be used except that the bandwidth demand value at the time of re-optimization will change while the network capacity will remain the same. The important issue to note is that if an LSP for an explicit route is to be released and a new one is to be established, some customers may be affected; therefore, minimizing this effect is important. However, bandwidth change on an already existing explicit route has little impact.
- Typically, the bandwidth requirement for customers is based on service level agreements (SLAs). Often, at any particular instant, the tunnels established may not be fully utilized by the customer, and/or if one customer is using them another customer may not be using them at the same time instant. Thus, a "bank"-style approach can also be taken. For example, a bank guarantees that it has the funds for your account; they do so in the hope that not all customers will withdraw all their money at the same time. A similar approach is possible in VPN networking. Suppose that we assume that each customer is likely to use about 80% of their bandwidth requirement on *average*. Then, this can be taken into account in LSP generations since RSVP-TE includes a controlled services option. This can be taken into account in the network modeling approach; for example, the 45-Mbps requirement of Customer A can be replaced by 36 Mbps ($= 0.8 \times 45$), and similarly for others. Accordingly, the link capacity constraints can be adjusted.

If the network is large and a large number of customers are to be supported, then the number of tunnels to be set up will also grow. Thus, the use of an automated configuration management system to invoke tunnel set-up would be required; such a management system can also check for label assignment and addresses mapping issues to ensure that different customers paths are assigned properly. Second, the number of candidate paths that is to be considered in the network flow modeling approach can be generated using the k-shortest path approach (refer to Section 2.10). The network modeling formulation for the general case is Model (4.5.3), presented in Chapter 4, and is thus not reproduced here.

Finally, while CPLEX is efficient in solving linear programming problems, it is time consuming to solve large *integer* linear programming problems due to their combinatorial nature. Thus, other specialized algorithms may be developed. A detailed discussion about such approaches can be found, for example, in [666].

23.2.4 LAYER 2 VPN TRAFFIC ENGINEERING

In layer 2 VPN, the CE device is a layer 2 device, not a router; we have discussed the basic concepts behind layer 2 VPN using MPLS in Section 22.5.2 to provide virtual private LAN service. We first

briefly explain again why such a service is appealing. Consider again a customer that has corporate offices in two different locations where their layer 2 facility is Ethernet based. This customer wants a connectivity to be set up between these two sites instead of assigning separate IP address blocks so that it appears as if it is part of the same LAN. This way, it can have a common *supernetted* subnet that covers both sites, and the entity to be shipped between the different sites is Ethernet frames. This approach of using Ethernet as the bearer is also appealing for carrying any protocol other than IP.

The question from the point of view of a layer 2 VPN provider is how to route such a layer 2 request between the customer's sites. For the VPN provider, there might be many such requests from different customers to facilitate. In each case, the customers enter a VPN network from a CE device to an ingress edge router in which lookup tables for LSPs to the ingress node must be configured. In fact, conceptually this picture is not different than the view shown in Figure 23.2(c). We can now assume Customer A to be the one wanting a layer 2 service between SF and KC; it is similar for other customers.

Thus, for the VPN provider to do traffic engineering based on many customers' requests, the basic network flow model is then the same as the one described in Section 23.2.3, which is a non-splittable multicommodity network flow (MCNF) model, for determining optimal LSP paths through the provider's network. In this case, the demand volume request can be based on the customer's own estimate of how much they need, which becomes the bandwidth request to the VPN provider, for example, in terms of an SLA.

Consider again Figure 23.2(c). Note that both customers B and C have demand requests between SF and NY. In this specific example, the routes through the network were found to be different based on the optimization goal. Suppose that the route selected were found to be the same, say they are both to use SF–KC–NY route. This brings an interesting question for the VPN provider—should the provider combine these demand requests into a single LSP on the route SF–KC–NY? From the traffic point of view, since there are two labels (refer to Section 22.5.2), it is certainly possible to differentiate traffic for different customers at the edge devices while using a common LSP for both. An advantage of combining such requests is that there are a fewer number of tunnels to manage and track within the VPN provider's network. On the other hand, if each customer has a different bandwidth request, it must be ensured at the ingress point that no individual customer's agreed upon bandwidth receives more than its share.

23.2.5 OBSERVATIONS AND GENERAL MODELING FRAMEWORK

From the illustration of different scenarios above for VPN traffic engineering, whether layer 2 or layer 3, we can say that the VPN routing/traffic engineering problem can be classified as a Type-B classification according to our service classification tabulated earlier in Table 21.1. Thus, this use of MPLS is primarily a transport network service mode. If such requests are to be set up on a semi-permanent basis, and different customer requests might arrive over a time horizon for tunneled services, then from the point of view of the VPN provider, the network traffic engineering problem can fall under the transport network routing framework, similar to optical networking discussed in Chapter 24. This means that this is a multitime period VPN transport routing problem to consider for the provider; see [666, Chapter 11] for a detailed discussion on multiperiod network design.

It is also possible that some customers might want protection and restoration of traffic engineering tunnels through a VPN provider's network. From an MPLS functionality point of view, FAST-

REROUTE can be used. From modeling the route selection for the primary and back-up paths for many such requests, while some might have partial protection, the model presented in Section 24.5 can be used.

Finally, under certain situations, dynamic transport and reconfigurability of LSP tunnels for customers are also permissible; if so, then a Type-C classification, listed in Table 21.1, is also applicable.

23.3 MULTICAST VPN TRAFFIC ENGINEERING

Now consider the case of point-to-multipoint virtual private LAN service (refer to Section 22.5.2). From the service provisioning point of view, the P2MP scenario would then require a tree structure for delivery within the MPLS networks. Such a tree structure can be addressed in the following ways: (1) set up multiple point-to-point LSP tunnels as before; the ingress router generates multiple copies to be sent on each LSP tunnel destinations, or (2) the MPLS has multicast functionality. How do we handle these from the point of view of traffic engineering by the VPN provider?

If multiple tunnels are to be done due to lack of P2MP capability in the MPLS VPN network, then differ requests for different sites to be identified first and then the point-to-point model from Section 23.2.3 can again be used for determining optimal tunnels.

If the network is equipped with multicast functionality, then for the P2MP case, a candidate tree generation concept instead of a candidate path generation concept in the MCNF model can be used. For, how to generate such candidate trees, see the k-shortest tree algorithm presented in [736], or the multicommodity network flow modeling framework discussed in [666, § 4.6.2].

We will explain this here by using the same example illustrated in Section 23.2.1 with a slight change. For Customer A, we assume that bandwidth requirements between all three sites is uniform at 50 Mbps. You will easily note from the demand matrix that is in Table 23.1 that Customer A can be served by an MP2MP LSP since this customer is connected to all three sites, now with the same bandwidth requirement.

There are three potentially multicast tree structures we can choose from: they can be identified as SF–KC–NY, SF–NY–KC, or KC–NY–SF. For this illustration, we ignore the direction as each MP2MP can be represented by an appropriate set of P2MP LSPs in actual deployment. As you may guess, instead of using a set of candidate paths to choose from, we can now use a set of candidate trees to choose from, each of which is labeled by the binary variables, `x_A_MP2MP_sf_kc_ny`, `x_A_MP2MP_sf_ny_kc`, and `x_A_MP2MP_kc_ny_sf`. Each of them may look like a path; but they are a tree in this instance. Since the decision problem needs to choose only one tree at optimality, then the following equation must be satisfied:

```
x_A_MP2MP_sf_kc_ny + x_A_MP2MP_kc_ny_sf + x_A_MP2MP_ny_sf_kc = 1
```

There is now another advantage of this approach. In the traffic engineering design phase, we can mix unicast and multicast LSP requirements together. To show, this consider that

```
Minimize x_A_MP2MP_sf_kc_ny + x_A_MP2MP_kc_ny_sf + x_A_MP2MP_ny_sf_kc
       + x_B_sf_ny + x_B_sf_kc_ny + x_C_sf_ny + x_C_sf_kc_ny
subject to
 d50MP2MP: x_A_MP2MP_sf_kc_ny + x_A_MP2MP_sf_ny_kc + x_A_MP2MP_kc_ny_sf = 1
```

```
 d80_B_sf_ny:   x_B_sf_ny + x_B_sf_kc_ny = 1
 d100_C_sf_ny:  x_C_sf_ny + x_C_sf_kc_ny = 1
 l_sf_kc: 50 x_A_MP2MP_sf_kc_ny + 50 x_A_MP2MP_ny_sf_kc
    + 80 x_B_sf_kc_ny + 100 x_C_sf_kc_ny <= 155
 l_sf_ny: 50 x_A_MP2MP_kc_ny_sf + 50 x_A_MP2MP_ny_sf_kc
    +  80 x_B_sf_ny + 100 x_C_sf_ny <= 155
 l_kc_ny: 50 x_A_MP2MP_sf_kc_ny + 50 x_A_MP2MP_kc_ny_sf
    +  80 x_B_sf_kc_ny + 100 x_C_sf_kc_ny <= 155
Integer
 x_A_MP2MP_sf_kc_ny  x_A_MP2MP_kc_ny_sf  x_A_MP2MP_ny_sf_kc
 x_B_sf_ny   x_B_sf_kc_ny
 x_C_sf_ny   x_C_sf_kc_ny
End
```

Certainly, to solve this optimization problem, you have to generate trees for each multicast request. This can be easily done using any of the tree generation algorithms to generate k-shortest trees, very much like using k-shortest paths for the list of candidate paths.

23.4 ROUTING/TRAFFIC ENGINEERING FOR VOICE OVER MPLS

Real-time interactive applications such as voice and multimedia can also be carried over MPLS. This means that for the duration of the call, a connection is set up for a voice call and MPLS then provides reserved paths for the voice call through an MPLS network, much like circuit-switching for packet delivery. The connection set-up aspect can be, for example, SIP-based; this will be discussed later in Section 26.4.3. In general, voice over MPLS can mean either (a) Voice over IP over MPLS, or (b) Voice directly over MPLS. Sometimes, Voice over ATM over MPLS is also listed under this category.

To directly do voice over MPLS, the basic idea used is to set up LSP tunnels as traffic trunks, and then multiplex multiple calls on the same LSP. Such LSPs may be sent up at an end-to-end basis with the MPLS network to carry a voice call, or a call may travel over multiple LSPs.

The MFA forum [579] has standardized the LSP structure for multiplexing voice calls, which is shown in Figure 23.3. An LSP has an outer label that identifies an LSP for two end points, and one or more VoMPLS primary subframes. Between the outer label and primary subframe an optional inner label is also allowed. Each primary subframe carries four fields: Channel ID, payload type, counter, and length. The channel ID field is to identify VoMPLS channels. Up to 248 channels can be multiplexed within a single LSP tunnel; however, using the inner label, the stacked label property of MPLS can be invoked to allow multiple different streams within an LSP.

You may note that VoMPLS falls under a Type-A classification, listed earlier in Table 21.1. This means that a call is to be established as soon as the request arrives, bandwidth is to be reserved on a label switch path for each voice call, and thus, link capacity resources are used on the link that an LSP traverses. On average, the duration of such a call is relatively short. Note that an LSP is *not* set for each call; rather, an LSP is set up to serve as trunkgroups between MPLS routers, on a periodic basis—thus, there is no scalability issue of setting up such LSPs using, say RSVP-TE. This is not to be confused with call set up signaling on a per call basis that can be based on either ISUP messaging or SIP.

In a sense, this usage of MPLS for voice service is essentially the same as the QoS routing discussed in greater depth in Chapter 21. We refer the reader to this chapter for further details. Note that here MPLS LSPs serve as trunkgroups between two routers, thus forming a virtual topology in which calls are to be routed. The alternate call routing concept discussed in Chapter 21 means that in this MPLS environment, the MPLS routers for voice services would need to have alternate call routing capabilities.

If alternate call routing functionality is not available, then LSP tunnels set up for trunkgroups would serve as direct links. The capacity of these links would need to be engineered so that call blocking probability is kept low at an acceptable grade-of-service. Voice traffic engineering for circuit-switching is discussed in Chapter 20, and for VoIP in Chapter 26.

An important issue is that call traffic volume can vary over time. Either LSP can be set up statically with plenty of bandwidth that does not change over time, or it may frequently be set up along the way with the required capacity to meet grade-of-service. Note that this may result in bandwidth allocation to an LSP, and then deallocation later since it is not needed due to a drop in call traffic volume.

In the presence of dynamic traffic, the dynamic allocation/deallocation problem can actually lead to network instability showing oscillatory behavior when there is no control. This oscillatory behavior is shown in Figure 23.4(a); here, to denote change in offered traffic over time, a sinusoidal traffic arrival curve is used that is subjected to allocation and deallocation of bandwidth on a tunnel if the blocking is below acceptable QoS tolerance or if it is above QoS tolerance, respectively. If simple controls such as hold-down timer between updates are used, it is possible to arrive at a stable environment (see Figure 23.4(b)). Thus, in a dynamic set up of LSPs, to meet service guarantee requirements, it is important to consider the LSP bandwidth update procedure in a way that avoids network instability. See [325] for further discussion.

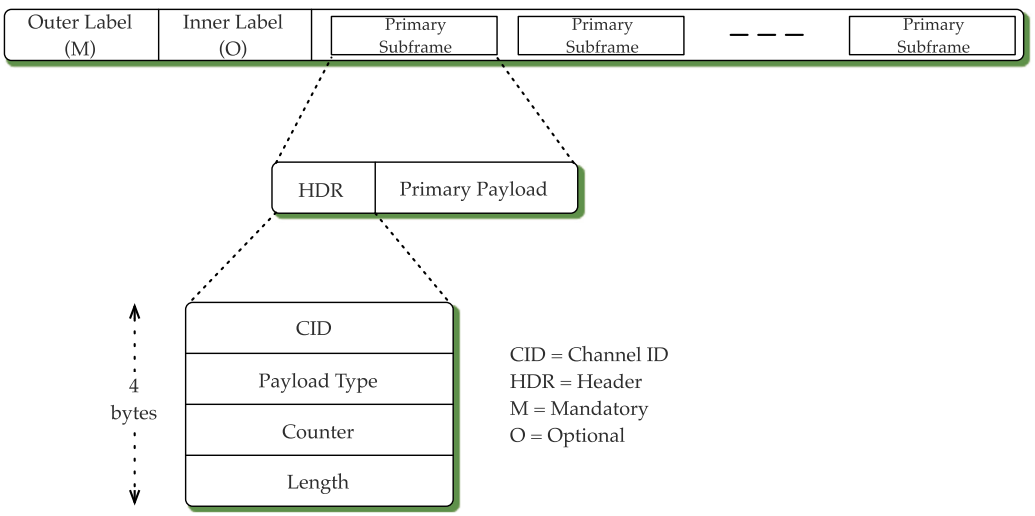


FIGURE 23.3

LSP Structure in VoMPLS (source: [579]).

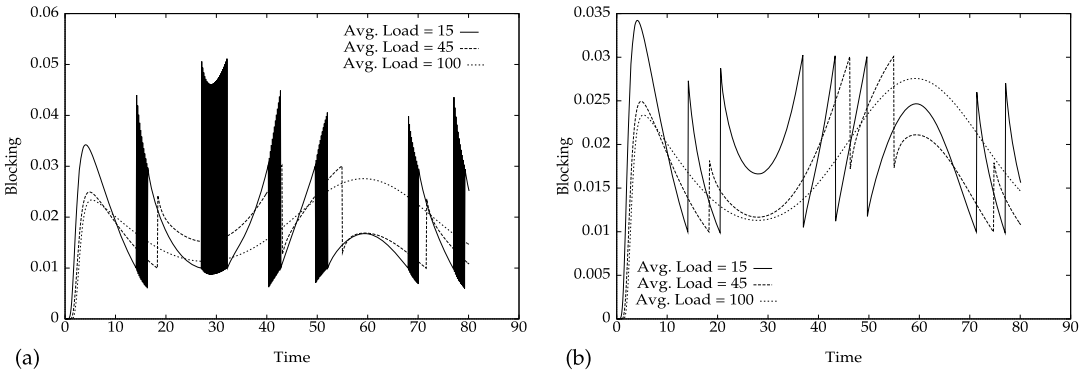


FIGURE 23.4

Transient performance due to an LSP bandwidth allocation/deallocation scheme: (a) instability; (b) corrected through control.

23.5 SUMMARY

In this chapter, we presented a set of routing and traffic engineering problems in which MPLS can be used. In general, an MPLS traffic engineering-based approach requires several issues to be considered such as path management, traffic assignment, network information dissemination, and network management [60]. Here, we have highlighted several approaches for traffic assignment and path management for different MPLS-based environments.

An important flexibility about MPLS is that depending on the service offered, it can fall in one of the three classifications identified earlier in Table 21.1. MPLS provides powerful capabilities if you know how to use it. In this chapter, we have presented a set of examples to illustrate the flexibility of MPLS. Furthermore, we have illustrated how various routing paradigms including a network flow-based modeling approach can be helpful in determining the optimal routing for a particular problem.

FURTHER LOOKUP

For a historical treatment of the "birth" and development of MPLS, including an excellent organization of Historical Internet drafts and RFCs on this subject, see [321].

For a provider provisioned VPN, refer to [610]. See [60] for a discussion on issues related to traffic engineering in an IP network; also, see [63]. For dynamic two-layer reconfigurability, see [565]. For an early implementation of MPLS-enabled switch for routing, see [144].

For a dynamic MPLS environment with anticipation of future service requests, a network would need to consider minimum interference routing [428], [883]. Such services can be served also using mechanisms that are variations of trunk reservation; the concept of trunk reservation is discussed elsewhere in this book.

Additional information about voice over MPLS can be found with the MFA forum; see [579].

EXERCISES

23.1 Discuss where and how MPLS-based IP/MPLS traffic engineering is different from "pure" IP traffic engineering.

23.2 Consider the network illustrated in Figure 23.2(a) and the network flow modeling approach described in Section 23.2.3.

- **(a)** Extend the model if the traffic trunk for customer C is allowed to be split. Determine the optimal flows and tunnels if the objective function is the same as the one discussed in Section 23.2.3.
- **(b)** Assume that customer B requires protection through FAST-REROUTE using a backup path. Extend the model to accommodate this change. Does the network have enough capacity to accommodate this request? If not, determine minimum additional capacity needed on each link if the objective is to load balance the network with no link having more than 70% utilization once this new capacity is added.
- **(c)** Each customer requests full protection back up tunnels with dedicated tunnels. Does the network have enough capacity to meet this request? If not, determine the minimum additional capacity needed in the network to serve this request, and determine the optimal LSP tunnel routing configuration.
- **(d)** Suppose that Customer-A wants a full-protection backup path, Customer-B wants a partially-protection backup up path with a 50% guarantee, and Customer-C requests a basic MPLS tunnel service with guarantee bandwidth, but no protection for failure. Present a network flow optimization model. Determine if any additional capacity is needed in the network; if so, determine the minimum additional capacity needed and the optimal LSP tunnel routing configuration.

23.3 The model in Section 23.2.3 is for minimum cost routing. Present a traffic engineering formulation if the objective is to load balance (congestion minimization).

23.4 Consider the model in Section 23.2.3. Suppose that you decided to aggregate the different customers' on a node-pair basis. Reformulate the traffic engineering problem. What are advantages and disadvantages of the aggregated model compared to the original model in Section 23.2.3.

23.5 Generalize the network flow model for traffic engineering for a network of L links and K demands in which K_1 customers require path protection, K_2 customers allow the traffic trunk to be split, and K_3 require no-split traffic trunks.

23.6 Generalize the multicast VPN traffic engineering problem by presenting a general formulation.