6

Core Capacity Planning and Traffic Engineering

This chapter addresses core capacity planning and how traffic engineering can be used as a tool to make more efficient use of network capacity.

6.1 Core Network Capacity Planning

Capacity planning of the core network is the process of ensuring that sufficient bandwidth is provisioned such that the committed core network SLA targets of delay, jitter, loss, and availability can be met. In the core network where link bandwidths are high and traffic is highly aggregated, the SLA requirements for a traffic class can be translated into bandwidth requirements, and the problem of SLA assurance can effectively be reduced to that of bandwidth provisioning. Hence, the ability to assure SLAs is dependent upon ensuring that core network bandwidth is adequately provisioned, which is in turn dependent upon core capacity planning.

The simplest core capacity planning processes use passive measurements of core link utilization statistics (i.e. as described in Chapter 5, Section 5.2) and apply rules of thumb, such as upgrading links when they reach 50% average utilization, or some other such general utilization target. The aim of such simple processes is to attempt to ensure that the core links are always significantly over-provisioned

This chapter has benefitted enormously from the input of Thomas Telkamp, Director of Network Consulting at Cariden Technologies, Inc. Thomas's work formed the basis of the capacity planning section.

relative to the offered average load, on the assumption that this will ensure that they are also sufficiently over-provisioned relative to the peak load, that congestion will not occur, and hence the SLA requirements will be met. There are, however, two significant consequences of such a simple approach. Firstly, without a network-wide understanding of the traffic demands, even an approach which upgrades links when they reach 50% average utilization may not be able to ensure that the links are still sufficiently provisioned when network element (e.g. link and node) failures occur, in order to ensure that the committed SLA targets continue to be met. Secondly, and conversely, rule of thumb approaches such as this may result in more capacity being provisioned than is actually needed.

Effective core capacity planning can overcome both of these issues. Effective core capacity planning requires a way of measuring the current network load, and a way of determining how much bandwidth should be provisioned relative to the measured load in order to achieve the committed SLAs. Hence, in this section we present a holistic methodology for capacity planning of the core network, which takes the core traffic demand matrix and the network topology into account to determine how much capacity is needed in the network, in order to meet the committed SLA requirements, taking network element failures into account if necessary, while minimizing the capacity and cost associated with over-provisioning.

The methodology presented in this section can be applied whether Diffserv is deployed in the core or not. Where Diffserv is not deployed, capacity planning is performed on aggregate. Where Diffserv is deployed, while the fundamental principles remain the same, capacity planning per traffic class is needed to ensure that class SLA targets are not violated.

6.1.1 Capacity Planning Methodology

We distinguish the following steps in the process of capacity planning:

1. Collect the core traffic demand matrices (either on aggregate or per class) and add traffic growth predictions to create a traffic demand forecast. This step is described in Section 6.1.2.

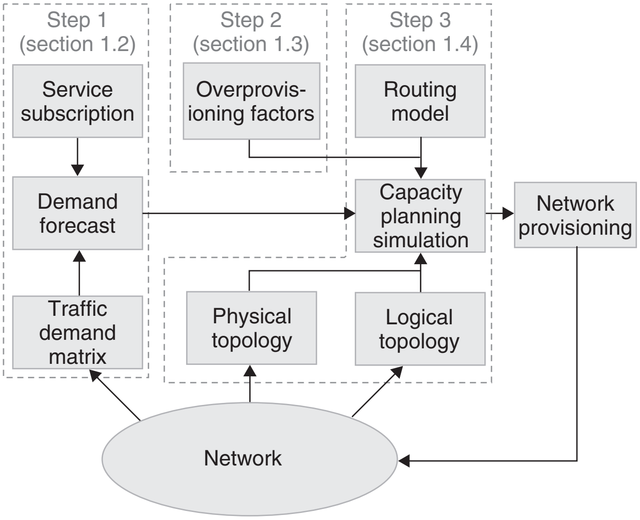


Figure 6.1 Capacity planning methodology

2. Determine the appropriate bandwidth over-provisioning factors (either on aggregate or per class) relative to the measured demand matrices, which are required to ensure that committed SLAs can be met. This step is described in Section 6.1.3.
3. Run simulations to overlay the forecasted demands onto the network topology, taking failure cases into account if necessary, to determine the forecasted link loadings. Analyze the results, comparing the forecasted link loadings against the provisioned bandwidth and taking the calculated over-provisioning factors into account, to determine the future capacity provisioning plan required to achieve the desired SLAs. This step is described in Section 6.1.2.

This capacity planning process is illustrated by Figure 6.1. The steps in the capacity planning process are described in detail in the proceeding sections.

6.1.2 Collecting the Traffic Demand Matrices

The core traffic demand matrix is the matrix of ingress to egress traffic demands across the core network. Traffic matrices can be measured or estimated to different levels of aggregation: by IP prefix, by router, by point of presence (POP), or by autonomous system (AS). The benefit of a core traffic matrix over simple per-link statistics is that the demand matrix can be used in conjunction with an understanding of the network routing model to predict the impact that demand growths can have and to simulate “what-if” scenarios, in order to understand the impact that the failure of core network elements can have on the (aggregate or per-class) utilization of the rest of the links in the network. With simple per-link statistics, when a link or node fails, in all but very simple topologies it may not be possible to know over which links the traffic impacted by the failure will be rerouted. Core network capacity is increasingly being provisioned taking single network element failure cases into account. To understand traffic rerouting in failure cases a traffic matrix is needed which aggregates traffic at the router-to-router level. If Diffserv is deployed, a per-class of service core traffic matrix is highly desirable.

The core traffic demand matrix can be an internal traffic matrix, i.e. router-to-router, or an external traffic matrix, i.e. router to AS, as illustrated in Figure 6.2, which shows the internal traffic demand

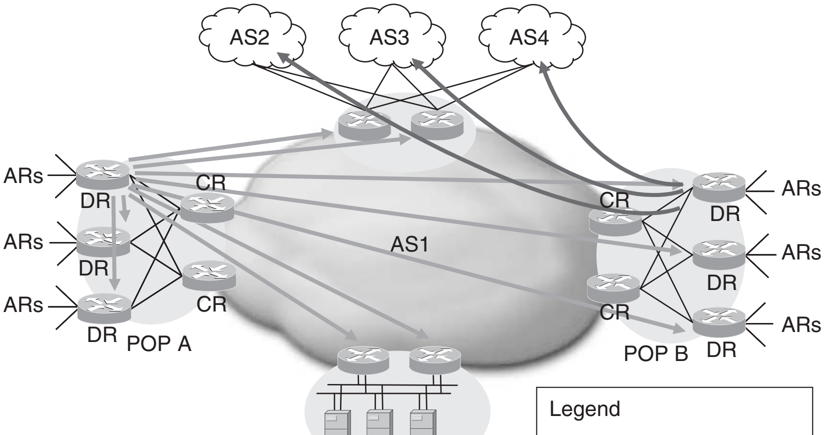


Figure 6.2 Internal and external traffic demands

matrix from one distribution router (DR), and the external traffic demand matrix from another.

The internal traffic matrix is useful for understanding the impact that internal network element failures will have on the traffic loading within the core. An internal matrix could also be edge-to-edge (e.g. DR to DR), or just across the inner core (e.g. CR to CR); a DR to DR matrix is preferred, as this can also be used to determine the impact of failures within a POP. The external traffic matrix provides additional context, which could be useful for managing peering connection capacity provision, and for understanding where internal network failures might impact in the external traffic matrix, due to closest-exit (a.k.a. "hot potato") routing.

There are a number of possible approaches for collecting the core traffic demand matrix statistics. The approaches differ in terms of their ability to provide an internal or external matrix, whether they can be applied to IP or MPLS, and whether they can provide a per-class of service traffic matrix. Further, the capabilities of network devices to provide information required to determine the core traffic matrix can vary depending upon the details of the particular vendor's implementation. Some of the possible approaches for determining the core traffic demand matrix are as follows:

- *IP flow statistics aggregation.* The Internet Protocol Flow Information eXport (IPFIX) protocol [IPFIX] is being defined within the IETF as a standard for the export of IP flow information from routers, probes, and other devices. If edge devices such as distribution routers are capable of accounting at a flow level (i.e. in terms of packet and byte counts), then a number of potential criteria could be used to aggregate this flow information – potentially locally on the device – in order to produce a traffic matrix.

  Where the Border Gateway Protocol (BGP) [RFC4271] is used within an AS, for example, each router at the edge of the AS is referred to as a BGP "peer." For each IP destination address that a peer advertises via BGP it also advertises a BGP next hop IP address, which is used when forwarding packets to that destination. In order to forward a packet to that destination, another BGP router within the AS needs to perform a recursive lookup, firstly looking in its BGP table to retrieve the BGP next hop address associated with that destination address, and then looking in its Interior Gateway Routing Protocol (IGP) routing table to determine how to get to that particular BGP next hop address (for further understanding on the workings of BGP, see [HALABI]). Hence, aggregating IPFIX flow statistics based upon the BGP next hop IP address used to reach a particular destination would produce an edge router to edge router traffic matrix.

- *MPLS LSP accounting.* Where MPLS is used, a label switch path (LSP) implicitly represents an aggregate traffic demand. Where BGP is deployed in conjunction with label distribution by the Label Distribution Protocol (LDP) [RFC3036], in the context of a BGP MPLS VPN service [RFC4364] for example, and each Provider Edge (PE) router[1] is a BGP peer, an LSP from one PE to another implicitly represents the PE-to-PE traffic demand. Hence, if traffic accounting statistics are maintained per LSP, these can be retrieved, using SNMP for example, to produce the PE-to-PE core traffic matrix.

  If MPLS traffic engineering is deployed (see Section 6.2.3) with a full mesh of TE tunnels, then each TE tunnel LSP implicitly represents the aggregate demand of traffic from the head-end router at the source of the tunnel, to the tail-end router at the tunnel destination. Hence, if traffic accounting statistics are maintained per TE tunnel LSP, these can be retrieved, using SNMP for example, to understand the core traffic matrix. If Diffserv-aware TE is deployed (see Section 6.2.3.2) with a full mesh of TE tunnels per class of service, the same technique could be used to retrieve a per-traffic class traffic matrix.

- *Demand estimation.* Demand estimation is the application of mathematical methods to measurements taken from the network, such as core link usage statistics, in order to infer the traffic demand matrix that generated those usage statistics. There are a number of methods that have been proposed for deriving traffic matrices from link measurements and other easily measured data [VARDI, TEBALDI, MEDINA, ZHANG], and there are a number of commercially available tools that use these, or similar, techniques in order to derive the core traffic demand matrix. If link statistics are available on a per-traffic class basis, then these techniques can be applied to estimate the per-class of service traffic matrix.

Further details on the options for deriving a core traffic matrix are provided in [TELKAMP1].

Whichever approach is used for determining the core traffic matrix, the next decision that needs to be made is how often to retrieve the measured statistics from the network. The retrieved statistics will normally be in the form of packet and byte counts, which can be used to determine the average traffic demands over the previous sampling interval. The longer the sampling interval, i.e. the less frequently the statistics are retrieved, the greater the possibility that significant variation in the traffic during the sampling interval may be hidden due to the effects of averaging. Conversely, the more frequently the statistics are retrieved, the greater the load on the system retrieving the data, the greater the load on the device being polled, and the greater the polling traffic on the network. Hence, in practice the frequency with which the statistics are retrieved is a balance, which depends upon the size of the network; in backbone networks it is common to collect these statistics every 5, 10, or 15 minutes.

The measured statistics can then be used to determine the traffic demand matrix during each interval. In order to make the subsequent stages of the process manageable, it may be necessary to select some traffic matrices from the collected data set. A number of possible selection criteria could be applied; one possible approach is to sum the individual (i.e. router to router) traffic demands within each interval, and to take the interval that has the greatest total traffic demand, i.e. the peak. Alternatively, in order to be sensitive to outliers (e.g. due to possible measurement errors), a high percentile interval such as the 95th percentile (P-95) could be taken, that is the interval for which more than 95% of the intervals have a lower value. In order to be representative, the total data set should be taken over at least a week, or preferably over a month, to ensure that trends in the traffic demand matrices are captured. In the case of a small network, it might be feasible to use all measurement intervals (e.g. all 288 daily measurements for 5-minute intervals), rather than to only use the peak (or percentile of peak) interval; this will give the most accurate simulation results for the network.

In geographically diverse networks, regional peaks in the traffic demand matrix may occur, such that most links in a specific region are near their daily maximum, at a time of the day when the total traffic in the network is not at its maximum. In a global network for example, in morning office hours in Europe, the European region may be busy, while the North American region is relatively lightly loaded. It is not very easy to detect regional peaks automatically, and one alternative approach is to define administrative capacity planning network regions (e.g. USA, Europe, Asia), and apply the previously described procedure per region, to give a selected per region traffic matrix.

Once the traffic matrix has been determined, other factors may need to be taken into account, such as anticipated traffic growth. Capacity planning will typically be performed looking sufficiently far in advance that new bandwidth could be provisioned before the network loading exceeds acceptable levels. If it takes 3 months to provision or upgrade a new core link, for example, and capacity planning is performed monthly, then the capacity planning process would need to try and predict at least 4 months in advance. If the expected network traffic growth within the next 4 months was 10%, for example, then the current traffic demand matrix would need to be multiplied with a factor of at least 1.1. Service subscription forecasts may be able to provide more granular predictions of future demand growth, possibly predicting the increase of particular traffic demands.

6.1.3 Determine Appropriate Over-provisioning Factors

The derived traffic matrices described in the previous section are averages taken over the sample interval, hence they lack information on the variation in traffic demands within each interval. There will invariably be bursts within the measurement interval that are above the average rate; if traffic bursts are sufficiently large temporary congestion may occur, causing delay, jitter, and loss, which may result in the violation of SLA commitments even though the link is on average not 100% utilized. To ensure that bursts above the average do not impact the SLAs, the actual bandwidth may need to be over-provisioned relative to the measure average rates. Hence, a key capacity planning consideration is to determine by how much bandwidth needs to be over-provisioned relative to the measured average rate, in order to meet a defined SLA target for delay, jitter, and loss; we define this as the over-provisioning factor (OP).

The over-provisioning factor required to achieve a particular SLA target depends upon the arrival distribution of the traffic on the link, and the link speed. Opinions remain divided on what arrival distribution describes traffic in IP networks. One view is that traffic is self-similar, which means that it is bursty on many or all timescales, i.e. whatever time period the traffic is measured over the variation in the average rate of the traffic stream is the same. An alternative view is that IP traffic arrivals follow a Poisson (or more generally Markovian) arrival process. For Poisson distributed traffic, the longer the time period over which the traffic stream is measured, the less variation there is in the average rate of the traffic stream. Conversely, the shorter the time interval over which the stream is measured, the greater the visibility of burst or the burstiness of the traffic stream. The differences in the resulting measured average utilization between self-similar and Poisson traffic, when measured over different timescales, are shown in Figure 6.3.

For Poisson traffic, queuing theory shows that as link speeds increase and traffic is more highly aggregated, queuing delays reduce for a given level of utilization. For self-similar traffic, however, if the traffic is truly bursty at all timescales, the queuing delay would not decrease with increased traffic aggregation. However, while views on whether IP network traffic tends toward self-similar [PAXON, SAHINOGLU], or Poisson [CAO, ZHANG] are still split, this does not fundamentally impact the capacity planning methodology we are describing. Rather, the impact of these observations is that, for high-speed links, the over-provisioning factor required to achieve a specified SLA target would

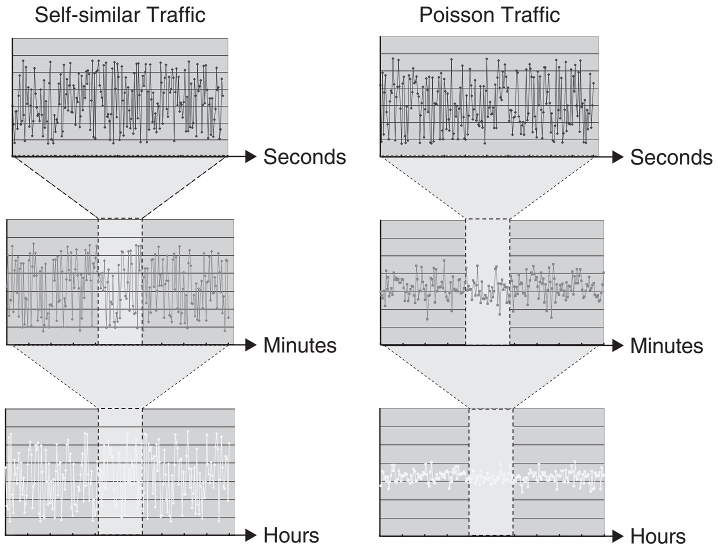


Figure 6.3 Self-similar versus Poisson traffic

need to be significantly greater for self-similar traffic, than for Poisson traffic.

Caveat Lector. A number of studies, both theoretical and empirical, have sought to quantify the bandwidth provisioning required to achieve a particular target for delay, jitter, and loss [FRALEIGH, BONALD, CHARNY, CAO, TELKAMP2], although none of these studies has yet been accepted as definitive. In the rest of this section, by way of example, we use the results attained in the study described in [TELKAMP2], to illustrate the capacity planning methodology. We chose these results because they probably represent the most widely used guidance with respect to core network over-provisioning.

In order to investigate bandwidth provisioning requirements, the authors of [TELKAMP2] captured a number of sets of packet level measurements from an operational IP backbone, carrying Internet and VPN traffic. The traces were used in simulation to determine the bursting and queuing of traffic at small timescales over this interval, to identify the relationship between measures of link utilization that can be easily obtained with capacity planning techniques (e.g. 5-minute average utilizations), and queuing delays experienced in much smaller timeframes,

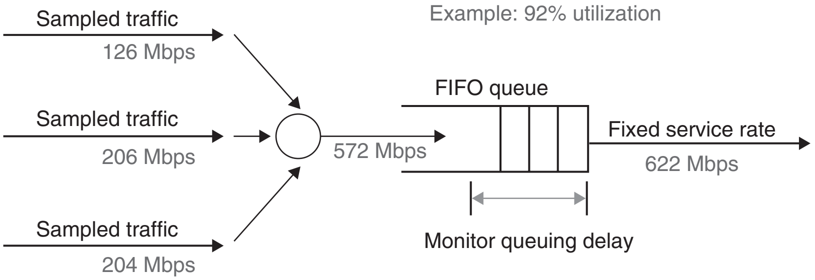


Figure 6.4 Queuing simulation from [TELKAMP2]

in order to determine the over-provisioning factors required to achieve various SLA targets. By using traces of actual traffic they avoided the need to make assumptions about the nature of the traffic distribution.

Each set of packet measurements or "trace" contained timestamps in microseconds of the arrival time for every packet on a link, over an interval of minutes. The traces, each of different average rates, were then used in a simulation where multiple traces were multiplexed together and the resulting trace was run through a simulated fixed speed queue, e.g. at 622 Mbps, as shown in Figure 6.4.

In the example in Figure 6.4, three traces with 5-minute average rates of 126 Mbps, 206 Mbps, and 240 Mbps respectively are multiplexed together resulting in a trace with a 5 minute average rate of 572 Mbps, which is run through a 622 Mbps queue, i.e. at a 5-minute average utilization of 92%. The queue depth was monitored during the simulation to determine how much queuing delay was experienced. This process was then repeated, with different mixes of traffic; as each mix had a different average utilization, multiple data points were produced for a specific interface speed.

After performing this process for multiple interface speeds, results were derived showing the relationship between average link utilization and the probability of queuing delay. The graph in Figure 6.5 uses the results of this study to show the relationship between the measured 5-minute average link utilization and queuing delay for a number of link speeds. The delay value shown is the P99.9 delay, meaning that 999 out of 1000 packets will have a delay caused by queuing which is lower than this value.

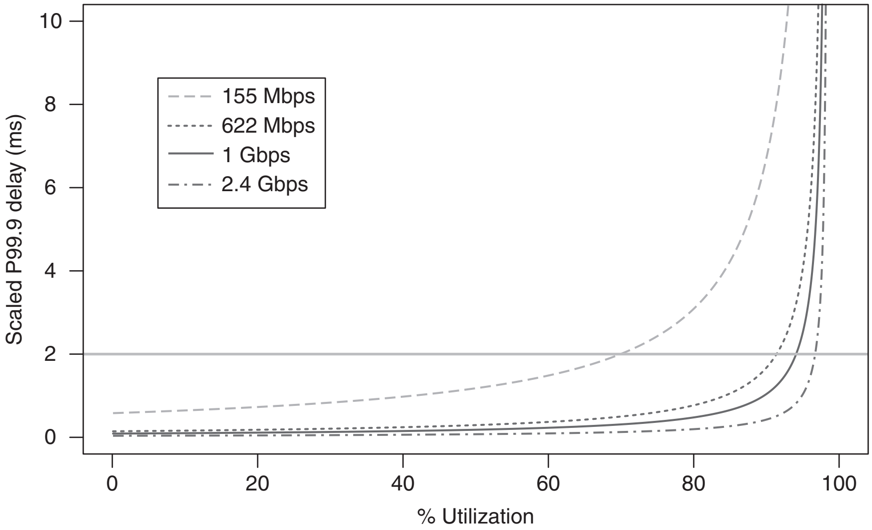


Figure 6.5 Queuing simulation results from [TELKAMP2]

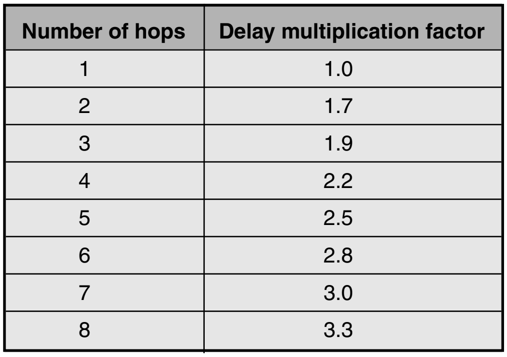
The x-axis in Figure 6.5 represents the 5-minute average link utilization; the y-axis represents the P99.9 delay. The lines show fitted functions to the simulation results for various link speeds, from 155 Mbps to 2.5 Gbps. Note that other relationships would result if the measured utilization was averaged over longer time periods, e.g. 10 minutes or 15 minutes, as in these cases there may be greater variations that are hidden by averaging, and hence lower average utilizations would be needed to achieve the same delay. The results in Figure 6.5 show that for the same relative levels of utilization, lower delays are experiences for 1 Gbps links than for 622 Mbps links, i.e. the level of over-provisioning required to achieve a particular delay target reduces as link bandwidth increases, which is indicative of Poisson traffic.

Taking these results as an example, we can use them to determine the over-provisioning factor that is required to achieve particular SLA objectives. For example, if we assume that Diffserv is not deployed in the core network and want to achieve a target P99.9 queuing delay of 2 ms on a 155 Mbps link, then from Figure 6.5, the 5-minute average link utilization should not be higher than approximately 70% or ~109 Mbps, i.e. an OP of 1/0.7 = 1.42 is required, meaning that the provisioned link bandwidth should be at least 1.42 times the 5-minute average link utilization. To achieve the same objective for a 1 Gbps link the 5-minute average utilization should be no more than ~96% or ~960 Mbps (i.e. OP = 1.04). Although the study from [TELKAMP2] did not focus on voice traffic, in similar studies by the same authors for VoIP-only traffic (with silence suppression) the OP factors required to achieve the same delay targets were similar.

We can apply the same principle on a per-class basis where Diffserv is deployed. To assure a P99.9 queuing delay of 1 ms for a class serviced with an assured forwarding (AF) PHB providing a minimum bandwidth assurance of 622 Mbps (i.e. 25% of a 2.5 Gbps link), the 5-minute average utilization for the class should not be higher than approximately 85% or ~529 Mbps. Considering another example, to assure a P99.9 queuing delay of 500 μs for a class serviced with an expedited forwarding (EF) per-hop behavior (PHB) implemented with a strict priority queue on a 2.5 Gbps link, as the scheduler servicing rate of the strict priority queue is 2.5 Gbps, the 5-minute average utilization for the class should not be higher than approximately 92% or ~2.3 Gbps (i.e. OP = 1.09) of the link rate. Note that these results are for queuing delay only and exclude the possible delay impact on EF traffic due to the scheduler and the interface FIFO as described in Chapter 2, Section 2.2.4.1.3.

The delay that has been discussed so far is *per-link* and not end-to-end across the core. In most cases, traffic will traverse multiple links in the network, and hence will potentially be subject to queuing delays multiple times. Based upon the results from [TELKAMP2], the P99.9 delay was not additive over multiple hops; rather, the table in Figure 6.6 shows the delay "multiplication factor" experienced over a number of hops, relative to the delay over a single hop.

If the delay objective across the core is known, the over-provisioning factor that needs to be maintained per-link can be determined. The core delay objective is divided by the multiplication factor from the table in Figure 6.6 to find the per-hop delay objective. This delay can then be looked up in the graphs in Figure 6.5 to find the maximum utilization for a specific link capacity that will meet this per-hop queuing delay objective. Consider for example, a network comprising



Number of hops	Delay multiplication factor
1	1.0
2	1.7
3	1.9
4	2.2
5	2.5
6	2.8
7	3.0
8	3.3

Figure 6.6 P99.9 delay multiplication factor

155 Mbps links with a P99.9 delay objective across the core network of 10 ms, and a maximum of 8 hops. From Figure 6.5, the 8 hops cause a multiplication of the per-link number by 3.3, so the per-link objective becomes 10 ms/3.3 = 3 ms. From Figure 6.6, the 3 ms line intersects with the 155 Mbps utilization curve at 80%. So the conclusion is that the 5-minute average utilization on the 155 Mbps links in the network should not be more than approximately 80% or ~124 Mbps (i.e. OP = 1.25) to achieve the goal of 10 ms delay across the core.

6.1.4 Simulation and Analysis

After obtaining the demand matrix, allowing for growth, and determining the over-provisioning factors required to achieve specific SLA targets, the final step in the capacity planning process is to overlay the traffic demands onto the network topology. This requires both an understanding of the network routing model – e.g. whether an interior gateway routing protocol (IGP), such as ISIS or OSPF, is used or whether MPLS traffic engineering is used – and an understanding of the logical network topology – i.e. link metrics and routing protocol areas – in order to understand the routing through the network that demands would take and hence to correctly map the demands to the topology. There are a number of commercially available tools, which can perform this function. Some such tools can also run failure case simulations, which consider the loading on the links in network element failures; it is common to model for single element failures, where an element could be a link, a node, or a shared risk link group (SRLG). SRLGs can be used to group together links that might fail simultaneously; to represent the failure of unprotected interfaces sharing a common linecard or circuits sharing a common fiber duct, for example. The concept of SRLGs can also be applied to more than just links, grouping links and nodes which may represent a shared risk, in order to consider what would happen to the network loading in the presence of the failure of a complete POP, for example.

The results of the simulation provide indications of the expected loading of the links in the network; this could be the aggregate loading or the per-class loading if Diffserv is deployed. The forecasted link loadings can then be compared against the provisioned link capacity, taking the calculated overprovisioning factors into account, to determine the future bandwidth provisioning plan required to achieve the desired SLAs. The capacity planner can then use this information to identify links which may be overloaded, such that SLAs will be violated, or areas where more capacity is provisioned than is actually needed.

6.2 IP Traffic Engineering

Capacity planning, as discussed in the proceeding section, is the process of ensuring that sufficient bandwidth is provisioned to assure that the committed core SLA targets can be met. IP traffic engineering is the logical process of manipulating traffic on an IP network to make better use of the network capacity, by making use of capacity that would otherwise be unused, for example. Hence, traffic engineering is a tool that can be used to ensure that the available network capacity is appropriately provisioned.

We contrast traffic engineering to network engineering, which is the physical process of manipulating a network to suit the traffic load, by putting in a new link between two POPs to support a traffic demand between them, for example. Clearly, network engineering and traffic engineering are linked; however, in this section we focus on the options for traffic engineering in an IP network. The outcome of the capacity planning process described in the previous section may drive the need for traffic engineering within a network.

In IP-based networks, traffic engineering is often considered synonymous with MPLS traffic engineering (TE) in particular, which is described in Section 6.2.3; however, there are other approaches in IP networks, including traffic engineering through the manipulation of Interior Gateway Routing Protocol (IGP) metrics – which is described in Section 6.2.2.

6.2.1 The Problem

In conventional IP networks IGPs such as OSPF [RFC2328] and IS-IS [RFC1142] forward IP packets on the shortest cost path toward the destination IP subnet address of each IP packet. The computation of the shortest cost path is based upon a simple additive metric (also known as weight or cost), where each link has an applied metric, and the cost for a path is the sum of the link metrics on the path. Availability of network resources, such as bandwidth, is not taken into account and, consequently, traffic can aggregate on the shortest (i.e. lowest cost) path, potentially causing links on the shortest path to be congested while links on alternative paths are under-utilized. This property of conventional IP routing protocols, of traffic aggregation on the shortest path, can cause suboptimal use of network resources, and can consequently impact the SLAs that can be offered, or require more network capacity than is optimally required.

Consider, for example, the network in Figure 6.7, where each link is 2.5 Gbps and each link has the same metric (assume a metric of 1). If there were a traffic demand of 1 Gbps from R1 to R8, and a traffic demand of 2 Gbps from R2 to R8, then the IGP would pick the same route for both traffic demands, i.e. R1/R2 → R3 → R4 → R7 → R8, because it has a metric of 4 (summing the metric of 1 for each of the links traversed) and hence is the shortest path.

Therefore, in this example, the decision to route both traffic demands by the top path (R3 → R4 → R7) may result in the path being

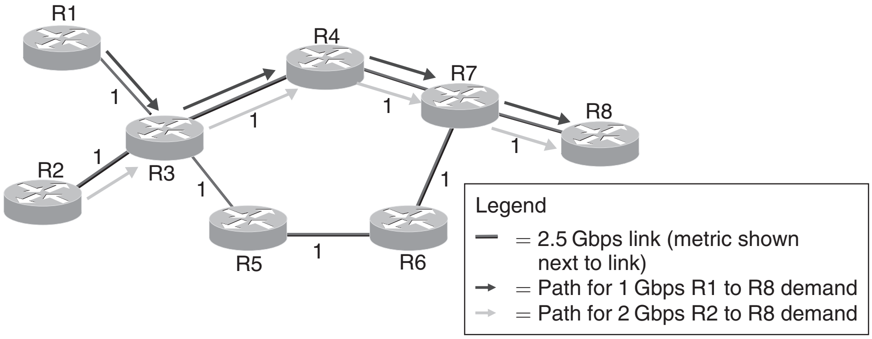

R1
R2
R3
R4
R5
R6
R7
R8
1
1
1
1
1
1
1
1
Legend
= 2.5 Gbps link (metric shown next to link)
= Path for 1 Gbps R1 to R8 demand
= Path for 2 Gbps R2 to R8 demand

Figure 6.7 Traffic engineering: the problem

congested, with a total offered load of 3 Gbps, while there is capacity available on the bottom path (R3 → R5 → R6 → R7). Traffic engineering aims to provide a solution to this problem.

The problem of traffic engineering can be defined as a mathematical optimization problem; that is, a computational problem in which the objective is to find the best of all possible solutions. Given a fixed network topology and a fixed source-to-destination traffic demand matrix to be carried, the optimization problem could be defined as determining the routing of flows that makes most effective use of (either aggregate or per-class) capacity. In order to solve this problem, however, it is important to define what is meant by the objective "most effective:" this could be to minimize the maximum link/class utilization in normal network working case conditions, i.e. when there are no network element failures. Alternatively the optimization objective could be to minimize the maximum link/class utilization under network element failure case conditions; typically single element (i.e. link, node, or SRLG) failure conditions are considered.

In considering the deployment of traffic engineering mechanisms, it is imperative that the primary optimization objective is defined, in order to understand what benefits the different options for traffic engineering can provide, and where traffic engineering will not help, but rather more bandwidth is required. Other optimization objectives are possible, such as minimizing propagation delay; however, if considered these are normally secondary objectives.

If we apply the primary optimization objective of minimizing the maximum link utilization in network working case (i.e. normal operating) conditions to the network shown in Figure 6.7 then the solution would be to route some subset of the traffic over the top path (R3 → R4 → R7) and the remainder over the bottom path (R3 → R5 → R6 → R7) such that congestion on the top path is prevented. If, however, we apply the primary optimization objective of minimizing the maximum link utilization during single network element failure case conditions, then on the failure of the link between R3 and R4, for example, both traffic demands R1 to R8 and R2 to R8 will be rerouted onto the bottom path (R3 → R5 → R6 → R7), which would be congested, as shown in Figure 6.8.

The example in Figure 6.8 is an illustration that traffic engineering cannot create capacity and that in some topologies, and possibly dependent upon the optimization objective, traffic engineering may not help. In network topologies that have only two paths available in normal network working case conditions, such as ring-based topologies, it is not possible to apply traffic engineering with a primary optimization objective of minimizing the maximum link utilization during network element failure case conditions; there is no scope for sophisticated traffic engineering decisions in network failure case conditions; if a link on one path fails, the other path is taken. In these cases, if congestion occurs during failure conditions then more capacity is simply required. More meshed

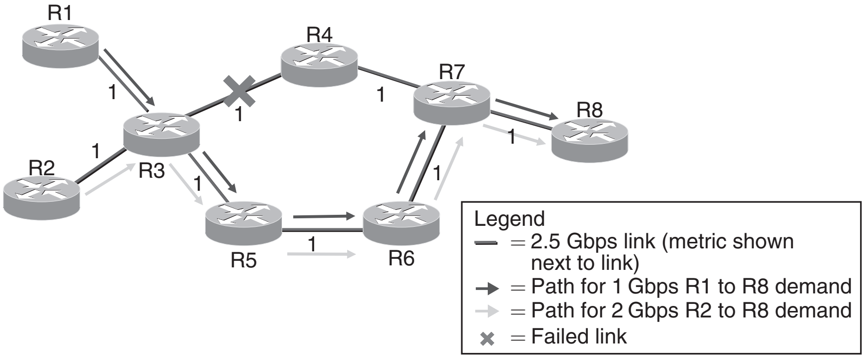

R1
R4
R7
R8
R2
R3
R5
R6
1
1
1
1
1
1
1
1
Legend
— = 2.5 Gbps link (metric shown next to link)
→ = Path for 1 Gbps R1 to R8 demand
→ = Path for 2 Gbps R2 to R8 demand
✖ = Failed link

Figure 6.8 Failure case optimization

network topologies may allow scope for traffic engineering in network element failure case conditions.

The chief benefit of traffic engineering is one of cost saving. Traffic engineering gives the network designer flexibility in how to manage their backbone bandwidth in order to achieve their SLAs. The more effective use of bandwidth potentially allows higher SLA targets to be offered with the existing backbone bandwidth. Alternatively, it offers the potential to achieve the existing SLA targets with less backbone bandwidth or to delay the time until bandwidth upgrades are required. The following conditions can all be drivers for the deployment of traffic engineering mechanisms:

- *Network asymmetry.* Asymmetrical network topologies can often lead to traffic being aggregated on the shortest path while other viable paths are under-utilized. Network designers will often try to ensure that networks are symmetrical such that where parallel paths exist, they are of equal cost and hence the load can be balanced across them using conventional IGPs, which support load balancing across multiple equal cost paths. Ensuring network symmetry, however, is not always possible due to economic or topological constraints; traffic engineering offers potential benefits in these cases.
- *Unexpected demands.* In the presence of unexpected traffic demands (e.g. due to some new popular content), there may not be enough capacity on the shortest path (or paths) to satisfy the demand. There may be capacity available on non-shortest paths, however, and hence traffic engineering can provide benefit.
- *Long bandwidth lead-times.* There may be instances when new traffic demands are expected and new capacity is required to satisfy the demand, but is not available in suitable timescales. In these cases, traffic engineering can be used to make use of available bandwidth on non-shortest path links.

The potential benefit of different approaches to traffic engineering can be quantified by using a holistic approach to capacity planning, such as described in Section 6.1, which is able to overlay the network traffic matrix on the network topology, while simulating the relative network loading taking into account different traffic engineering schemes. A network-by-network analysis is required to determine whether the potential TE benefit will justify the additional deployment and operational cost associated with the deployment of these technologies.

Traffic engineering can potentially be performed at layer 2 (i.e. by traffic engineering the underlying transport infrastructure) or at layer 3. In focussing on layer 3, in the following sections we consider possible approaches for IP traffic engineering, and consider traffic engineering at layer 2 to be an inception of network engineering when considered from a layer 3 perspective.

6.2.2 IGP Metric-based Traffic Engineering

The tactical and ad hoc tweaking of IGP metrics to change the routing of traffic and relieve congested hotspots has long been practiced in IP backbone networks. For a long time, however, this approach was not considered viable for systematic network-wide traffic engineering and it was often cited that changing the link metrics just moves the problem of congestion around the network. If we consider the network from Figure 6.7, by changing the metric of the link from R3 to R4 from 1 to 3, as can be seen in Figure 6.9, the traffic demands both from R1 to R and from R2 to R8 are now routed over the bottom path

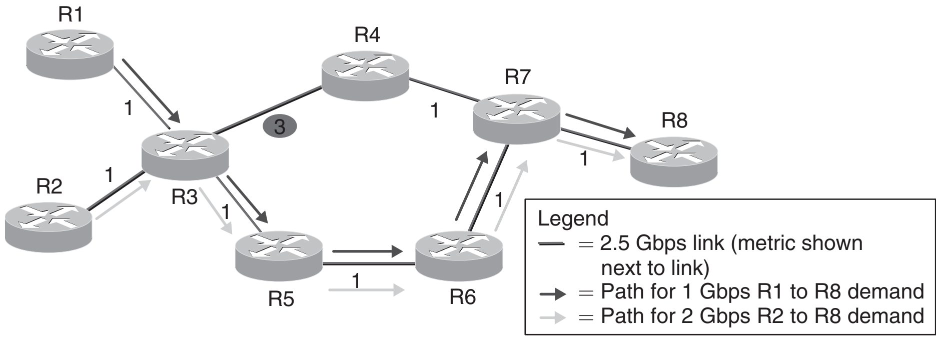

R1
R4
R7
R8
R2
R3
R5
R6
1
3
1
1
1
1
1
1
1
Legend
— = 2.5 Gbps link (metric shown next to link)
→ = Path for 1 Gbps R1 to R8 demand
→ = Path for 2 Gbps R2 to R8 demand

Figure 6.9 Changing link metrics moves congestion

(R3 → R5 → R6 → R7), which is now the least cost path (cost of 5). In this case the congestion has moved to the bottom path.

If instead, the metric of the link from R3 to R4 was changed from 1 to 2 (rather than 1 to 3), however, then the top path (R3 → R4 → R7) and the bottom path (R3 → R5 → R6 → R7) would have equal path costs of 5, as shown in Figure 6.10.

Where equal cost IGP paths exist, equal costs multipath (ECMP) algorithms are used to balance the load across the equal cost paths. There are no standards defining how ECMP algorithms should balance traffic across equal cost paths and different vendors may implement different algorithms. ECMP algorithms typically, however, perform a hash function on fields in the header of the received IP packets to determine which one of the paths should be used for a particular packet. A common approach is to perform the hash function using the 5-tuple of IP protocol, source IP address, destination IP address, source UDP/TCP port, and destination UDP/TCP as inputs. The result of such a hash function is that load balancing across equal cost paths would be achieved for general distributions of IP addresses and ports. Such approaches also ensure that packets within a single flow are consistently hashed to the same path, which is important to prevent resequencing within a flow due to the adverse impact that packet re-ordering can have on the performance of some applications (this is discussed in more detail in Chapter 1, Section 1.2.5).

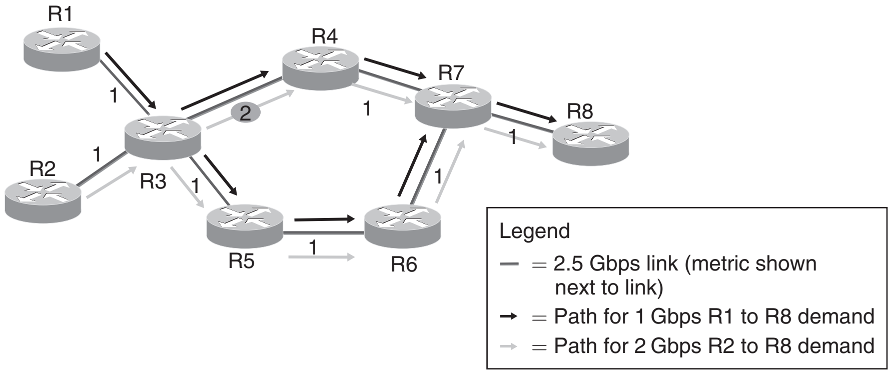

R1
R4
R7
R8
1
2
1
1
R2
1
R3
1
1
R5
1
R6
Legend
— = 2.5 Gbps link (metric shown next to link)
→ = Path for 1 Gbps R1 to R8 demand
→ = Path for 2 Gbps R2 to R8 demand

Figure 6.10 Equal IGP path costs

If such an ECMP algorithm were used in the example shown in Figure 6.10, and assuming a general distribution of addresses and ports, the 3 Gbps aggregate demand from R1 and R2 to R8, would be evenly distributed with approximately 1.5 Gbps on the top path and approximately 1.5 Gbps on the bottom path, and therefore the bandwidth would be used effectively and congestion would be avoided. Hence, the mantra that tweaking IGP metrics just moves the problem of congestion around the network is a generalization that is not always true in practice. For some symmetrical network topologies and matrices of traffic, ECMP algorithms may be able to distribute the load effectively without the need for other traffic engineering approaches at all.

In recognition of the possible application of metric-based traffic engineering, there has been a significant recent increase in research in the approach of systematic (i.e. network-wide) traffic engineering by manipulating IGP metrics [FORTZ1, LORENZ, BURIOL, ERICSSON, AMEUR]. Further, IGP metric-based traffic engineering has been realized in the development of automated planning tools, which take inputs of the network logical (i.e. IGP) and physical topology, together with the network traffic demand matrix and derive a more optimal set of link metrics based upon a defined optimization goal. These optimization goals may be to minimize the maximum utilization on aggregate, or per-class.

IGP metric-based traffic engineering provides less granular traffic control capabilities than MPLS traffic engineering (see Section 6.2.3). The effectiveness of IGP metric-based traffic engineering is dependent upon the network topology, the traffic demand matrix, and the optimization goal. [FORTZ2] shows that, for the proposed AT&T WorldNet backbone, they found weight settings that performed within a few percent of the optimal general routing, which is where the flow for each demand is optimally distributed[2] over all paths between source and destination. Studies by [GOUS] conclude that in the six networks they study, metric-based TE can be ~80–90% as efficient as the theoretical optimal general routing. Further, they surmise that the greatest relative difference in performance between IGP metric-based traffic engineering and traffic engineering via explicit routing (such as provided by MPLS traffic engineering) occurs in large networks with heterogeneous link speeds, i.e. where ECMP cannot be readily used to split traffic between parallel circuits with different capacities.

6.2.3 MPLS Traffic Engineering

Unlike conventional IP routing, which uses pure destination-based forwarding, multiprotocol label switching (MPLS), traffic engineering (TE) uses the implicit MPLS characteristic of separation between the data plane (also known as the forwarding plane) and the control plane to allow routing decisions to be made on criteria other than the destination address in the IP packet header, such as available link bandwidth. MPLS TE provides constraint-based path computation and explicit routing capabilities at layer 3, which can be used to divert traffic away from congested parts of the network to links where bandwidth is available and hence make more optimal use of available capacity. Label switched paths (LSPs), which are termed "traffic engineering tunnels" in the context of MPLS TE, are used to steer traffic through the network allowing links to be used which are not on the IGP shortest path to the destination.

It is noted that, as well as being used to solve the traffic engineering problem, MPLS TE has other applications including admission control (as described in Chapter 4, Section 4.4.6), route pinning,[3] and MPLS TE Fast Reroute (see Chapter 2, Section 2.6).

6.2.3.1 MPLS TE Example Tunnel Establishment

Consider the network in Figure 6.11, where every link is 2.5 Gbps and each has the same metric (assume a metric of 1), and where a single MPLS TE tunnel of 1 Gbps is already established from LSR1 to LSR8, using the path LSR1 → LSR3 → LSR4 → LSR7 → LSR8, because it is the shortest path (path cost = 4) with available bandwidth. In this example, it is assumed that the entire network has been enabled for MPLS TE, and that the full bandwidth on each interface is used for MPLS TE.

The following example sequence of events considers the establishment of another TE tunnel, a 2 Gbps tunnel from LSR2 to LSR8.

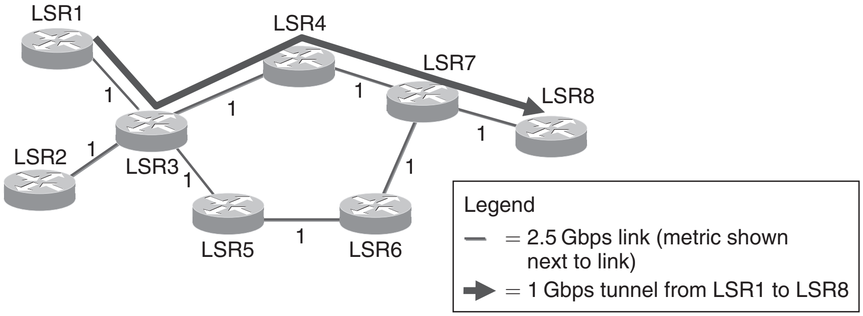

LSR1
LSR4
LSR7
LSR8
LSR2
LSR3
LSR5
LSR6
1
1
1
1
1
1
1
1
1
Legend
— = 2.5 Gbps link (metric shown next to link)
➡ = 1 Gbps tunnel from LSR1 to LSR8

Figure 6.11 MPLS TE example tunnel establishment

1. *Resource/policy information distribution*. Each router within the network floods information on the available bandwidth resources for its connected links, together with administrative policy constraint information, throughout the network by means of extensions to link-state based IGP routing protocols such as IS-IS [RFC3784] and OSPF [RFC3630].

   As TE tunnels are unidirectional, each TE-enabled router maintains a pool of available (i.e. currently unused) TE bandwidth in the egress direction for each interface that it has. Considering LSR3 for example, because the tunnel from LSR1 to LSR8 has already reserved 1 Gbps of bandwidth on the interface to LSR4, LSR3 will only advertise 1.5 Gbps worth of available bandwidth for that interface. For all of its other interfaces, LSR3 will advertise 2.5 Gbps of available bandwidth.

2. *Constraint-based path computation*. All of the routers within the MPLS TE area will receive the information on the available network resources, advertised via IS-IS or OSPF. With MPLS TE, tunnel paths can be specified manually, but more commonly are either dynamically calculated online in a distributed fashion by the TE tunnel sources (known as tunnel "*head-ends*") themselves or determined by an offline centralized function (also know as a tunnel server or path computation element) which then specifies the explicit tunnel path a head-end should use for a particular tunnel. With either approach, constraint-based routing is performed using

a constraint-based shortest path first (CSPF) algorithm to determine the path that a particular tunnel will take based upon a fit between the available network bandwidth resources (and optionally policy constraints) and the required bandwidth (and policies) for that tunnel. This CSPF algorithm is similar to a conventional IGP shortest path first (SPF) algorithm, but also takes into account bandwidth and administrative constraints, pruning links from the topology if they advertised insufficient resources, i.e. not enough bandwidth for the tunnel, or if they violate tunnel policy constraints. The shortest (i.e. lowest cost) path is then selected from the remaining topology. Whether online or offline path calculation is used, the output is an explicit route object (ERO) which defines the hop-by-hop path the tunnel should take and which is handed over to RSVP in order to signal the tunnel label switched path (LSP).

We assume online path calculation by the tunnel head-end, in this case LSR2. There are two possible paths from LSR2 to LSR8, either the top path (LSR2 → LSR3 → LSR4 → LSR7 → LSR8) or the bottom path (LSR2 → LSR3 → LSR5 → LSR6 → LSR7 → LSR8). As the tunnel from LSR2 to LSR8 is for 2 Gbps, there is insufficient bandwidth currently available (1.5 Gbps only) on the links from LSR3 → LSR4 and from LSR4 → LSR7 and hence the top path is discounted by the CSPF algorithm. Therefore, in this example the bottom path is the only possible path for the tunnel from LSR2 to LSR8, and output of the CSPF algorithm is an ERO which specifies the IP addresses of the hops on the path, i.e. LSR2 → LSR3 → LSR5 → LSR6 → LSR7 → LSR8.

3. *RSVP for tunnel signaling*. The Resource ReSerVation Protocol (RSVP) [RFC2205], with enhancements for MPLS TE [RFC3209], is used to signal the TE tunnel. RSVP is used differently in the context of MPLS TE than it is for per flow admission control, as described in Chapter 4, Section 4.4.1.

   RSVP uses two signaling messages, a Path message and a Resv message.

   i. The Path message carries the ERO and other information including the requested bandwidth for the tunnel, which is

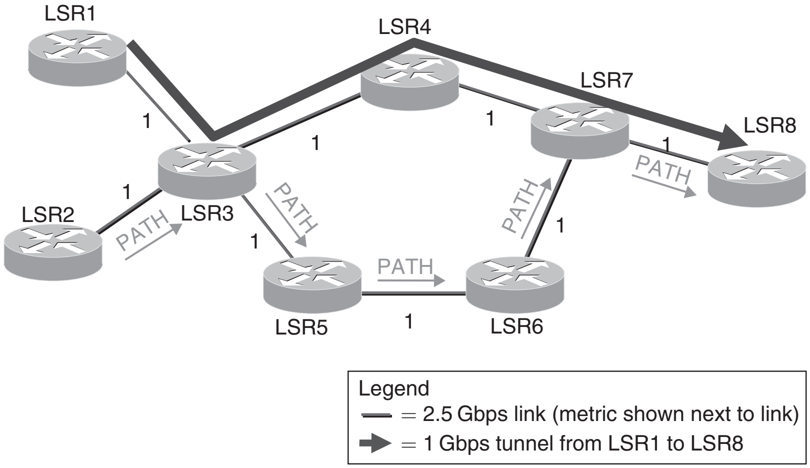

LSR1
LSR4
LSR7
LSR8
LSR2
LSR3
LSR5
LSR6
1
1
1
1
1
1
1
1
PATH
PATH
PATH
PATH
PATH
Legend
= 2.5 Gbps link (metric shown next to link)
= 1 Gbps tunnel from LSR1 to LSR8

Figure 6.12 MPLS TE example tunnel establishment: Step 3a

used for admission control. An RSVP Path message is sent from the tunnel head-end to the tunnel tail-end, as shown in Figure 6.12, explicitly routed hop-by-hop using the ERO.

At each router that receives the Path message an admission control decision is made to verify that the outbound interface that will be used to forward the Path message to the next hop defined by the ERO, has sufficient resources available to accept the requested bandwidth for the tunnel. This admission control decision may seem redundant as the CSPF algorithm has already picked a path with sufficient bandwidth; however, it is required because it is possible that the head-end router may have performed the CSPF algorithm on information which is now out of date, for example, if another tunnel has been set up in the intervening period since the tunnel path was calculated.

If the admission control decision is successful, the path message is forwarded to the next hop defined by the ERO, until the path message reaches the tail-end router. MPLS TE supports the concept of pre-emption and a lower priority tunnel may be pre-empted to allow a higher priority tunnel to be set up. If the admission control

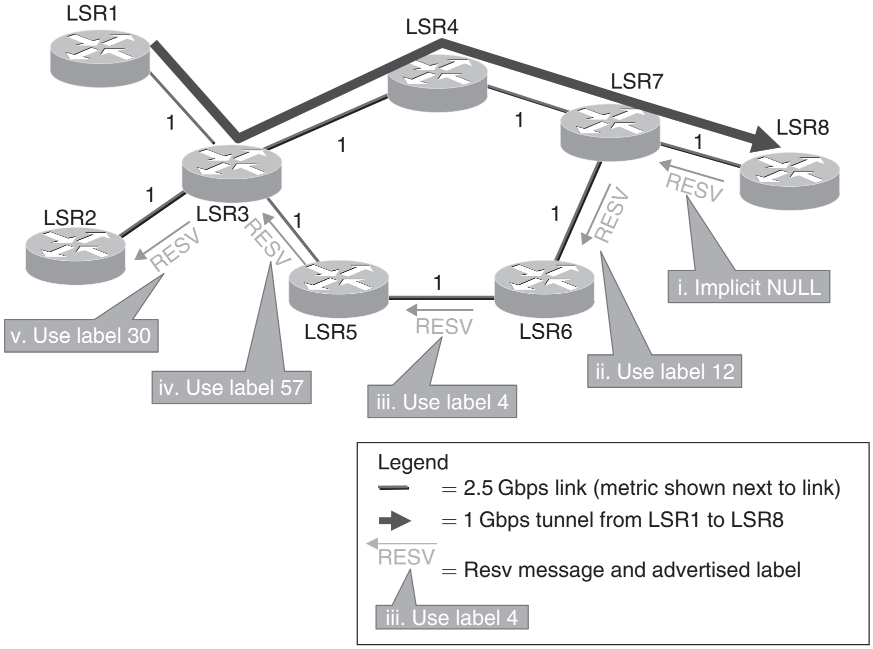


Figure 6.13 MPLS TE example tunnel establishment: Step 3b – label advertisement

decision is unsuccessful at any hop, a PathErr message is returned to the tunnel head-end.

It is noted that where RSVP is used for per flow admission control, rather than for MPLS TE tunnel signaling, the admission control decision is made in response to the receipt of the Resv message.

If the tail-end receives the Path message, then the admission control decisions must have been successful at each hop on the tunnel path. In response, the tail-end router originates a reservation (Resv) message which follows the path defined by the ERO in reverse in order to establish the LSP that defines the tunnel, as shown in Figure 6.13.

At each hop on the tunnel path that receives the Resv message, the tunnel reservation is confirmed. In order to set up the tunnel LSP, the Resv message is then forwarded to the upstream (i.e. closer to head-end) neighbor on the tunnel path, together with MPLS label value that this router expects to be used for traffic on the tunnel received from the upstream neighbor.

In this example, penultimate hop popping (PHP) is assumed and LSR8, as the final hop on the tunnel path, advertises an implicit null label to LSR7 accordingly. LSR7 then advertises label value 12 to LSR6, and so on, until the Resv message reaches the tunnel head-end. This is an example of downstream on demand label binding with upstream label distribution, where upstream/downstream is with reference to the direction of the flow packets on the LSP.

4. *Assigning traffic to tunnels*. When the Resv message reaches the head-end, the tunnel LSP has been successfully established and it can be used for traffic forwarding. There are a number of ways to determine when traffic should use the TE tunnel rather than the conventional IGP path. The simplest is to use static routing with a static route defining that traffic to a particular destination subnet address should use the tunnel rather than the conventional IGP route. Some vendors also support the capability to automatically calculate IP routes to forward traffic over MPLS TE tunnels, by adapting Dijkstra's SPF algorithm as described in [RFC3906].

   Having decided to forward some traffic onto the tunnel, the head-end router, in this case LSR2 assigns traffic to that tunnel by forwarding it on the tunnel LSP. It forwards traffic on the TE tunnel by sending it toward LSR3 with label value 30 as shown in Figure 6.14.

   LSR3 receives the labeled packet, and label switches it to LSR5 swapping the label from 30 to 57. Note that LSR3 uses only the label to determine how to forward the packet, i.e. it does not look at the underlying IP destination address. The tunneled packet continues on the LSP until it reaches LSR7, which as the penultimate hop, pops off the outer label and forwards it to LSR8, which is the tunnel tail-end. If a label stack is not used, the tail-end router looks at the IP destination address to determine how to forward the received packet; if a label stack is used (e.g. in the context of BGP MPLS VPNs as per RFC4364), the tail-end router uses the

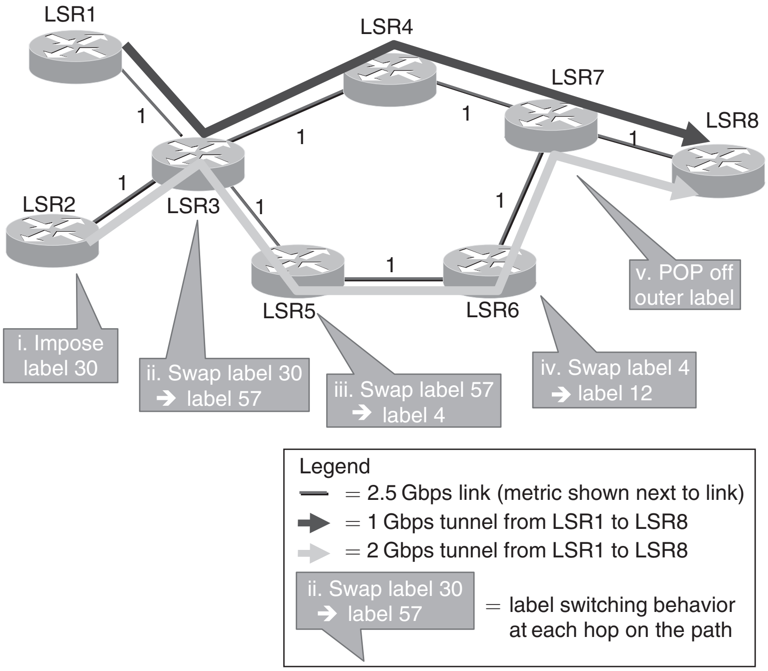

LSR1
LSR4
LSR7
LSR8
LSR2
LSR3
LSR5
LSR6
1
1
1
1
1
1
1
1
1
i. Impose label 30
ii. Swap label 30 ➔ label 57
iii. Swap label 57 ➔ label 4
iv. Swap label 4 ➔ label 12
v. POP off outer label
Legend
= 2.5 Gbps link (metric shown next to link)
= 1 Gbps tunnel from LSR1 to LSR8
= 2 Gbps tunnel from LSR1 to LSR8
ii. Swap label 30 ➔ label 57
= label switching behavior at each hop on the path

Figure 6.14 MPLS TE example tunnel establishment: Step 4 – label switching

outermost of the remaining labels to determine how to forward the received packet.

5. *TE tunnel control and maintenance*. Periodic RSVP Path/Resv messages maintain the tunnel state. Unlike tunnel setup, Path/Resv messages used for tunnel maintenance are sent independently and asynchronously.

The tunnel head-end can tear down a tunnel by sending a PathTear message. If a network element (link or node) on the tunnel path should fail, the adjacent upstream neighboring router on the tunnel path will send a PathErr message to the head-end, which will then attempt to recalculate a new tunnel path around the failed element. Similarly, if a tunnel is pre-empted, a PathErr message will be sent to the head-end, which will then attempt to recalculate a new tunnel path where bandwidth is available.

6.2.3.2 Diffserv-aware MPLS Traffic Engineering

MPLS TE and Diffserv can be deployed concurrently in an IP backbone, with TE determining the path that traffic takes on aggregate based upon aggregate bandwidth constraints, and Diffserv mechanisms being used on each link for differential scheduling of packets on a per-class of service basis. TE and Diffserv are orthogonal technologies which can be used in concert for combined benefit: TE allows distribution of traffic on non-shortest paths for more efficient use of available bandwidth, while Diffserv allows SLA differentiation on a per-class basis. As it was initially defined and has been described in the previous section, however, MPLS TE computes tunnel paths for aggregates across all traffic classes and hence traffic from different classes may use the same TE tunnels. In this form MPLS TE is aware of only a single aggregate pool of available bandwidth per link and is unaware of what specific link bandwidth resources are allocated to which queues, and hence to which classes.

Diffserv-aware MPLS TE (DS-TE) extends the basic capabilities of TE to allow constraint-based path computation, explicit routing and admission control to be performed separately for different classes of service. DS-TE provides the capability to enforce different bandwidth constraints for different classes of traffic through the addition of more pools of available bandwidth on each link. These bandwidth pools are sub-pools of the aggregate TE bandwidth constraint, i.e. the sub-pools are a portion of the aggregate pool. This allows a bandwidth sub-pool to be used for a particular class of traffic, such that constraint-based routing and admission control can be performed for tunnels carrying traffic of that class, with the aggregate pool used to enforce an aggregate constraint across all classes of traffic. There are two different models that define how the sub-pool bandwidth constraints are applied:

- *Maximum allocation model.* [RFC4127] defines the maximum allocation bandwidth constraints model (MAM) for Diffserv-aware MPLS TE. With the MAM, independent sub-pool constraints can be applied to each class, and an aggregate constraint can be applied across all classes.

- *Russian doll model*. [RFC4125] defines the Russian dolls bandwidth constraints model (RDM) for Diffserv-aware MPLS TE. With the RDM, a hierarchy of constraints is defined, which consists of an aggregate constraint (global pool), and a number of sub-constraints (sub-pools) where constraint 1 is a sub-pool of constraint 0, constraint 2 is a sub-pool of constraint 1, and so on.

The choice of which bandwidth allocation model to use depends upon the way in which bandwidth allocation and pre-emption will be managed between tunnels of different classes. It is noted that if traffic engineering is required for only one of the deployed traffic classes, e.g. for EF traffic only, then DS-TE is not required and standard single bandwidth pool TE is sufficient.

In support of DS-TE, extensions have been added to IS-IS and OSPF [RFC4124] to advertise the available sub-pool bandwidth per link. In addition, the TE constraint-based routing algorithms have been enhanced for DS-TE in order to take into account the constraint of available sub-pool bandwidth in computing the path of sub-pool tunnels. RSVP has also been extended [RFC4124] to indicate the constraint model and the bandwidth pool, for which a tunnel is being signaled.

As described in Section 6.1.3, setting an upper bound on the EF class (e.g. VoIP) utilization per link is necessary to bound the delay for that class and therefore to ensure that the SLA can be met. DS-TE can be used to assure that this upper bound is not exceeded. For example, consider the network in Figure 6.15, where each link is 2.5 Gbps and an IGP and TE metric value of one is applied to each link.

DS-TE could be used to ensure that traffic is routed over the network so that, on every link, there is never more than a defined percentage of the link capacity for EF class traffic, while there can be up to 100% of the link capacity for EF and AF class traffic in total. In this example, for illustration we assume that the defined maximum percentage for EF traffic per link is 50%. LSR1 is sending an aggregate

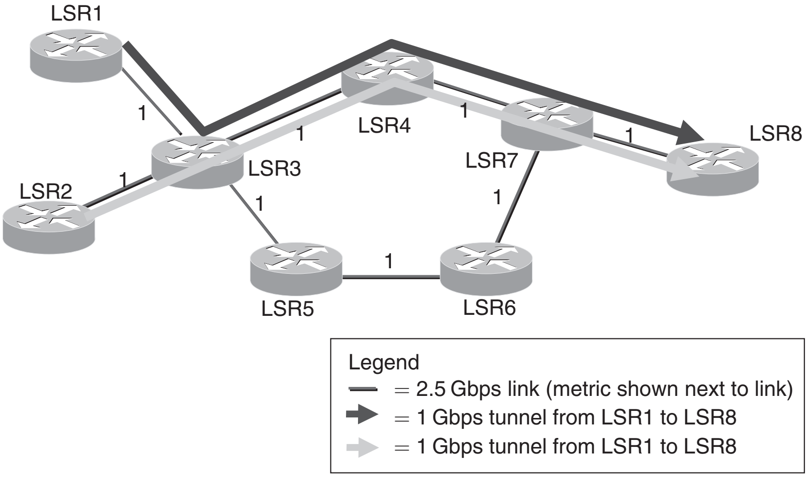

LSR1
LSR2
LSR3
LSR4
LSR5
LSR6
LSR7
LSR8
1
1
1
1
1
1
1
1
Legend
= 2.5 Gbps link (metric shown next to link)
= 1 Gbps tunnel from LSR1 to LSR8
= 1 Gbps tunnel from LSR1 to LSR8

Figure 6.15 DS-TE deployment example 1

of 1 Gbps of traffic to LSR8, and R2 is also sending an aggregate of 1 Gbps of traffic to LSR8. In this case, both the IGP (i.e. if TE were not deployed) and non-Diffserv aware TE would pick the same route. The IGP would pick the top route (R1/R2 → R3 → R4 → R5 → R8) because it is the shortest path (with a metric of 4). Assuming 1 Gbps tunnels were used from both LSR1 and LSR2 to LSR8, TE would also pick the top route, because it is the shortest path that has sufficient bandwidth available (metric of 4, 2.5 Gbps bandwidth available, 2 Gbps required). The decision to route both traffic aggregates via the top path may not seem appropriate if we examine the composition of the aggregate traffic flows.

If each of the aggregate flows were composed of 250 Mbps of VoIP traffic and 750 Mbps of standard data traffic, then in this case the total VoIP traffic load on the top links would be 500 Mbps, which is within our EF class per link bound of 50% = 1 Gbps. If, however, each traffic aggregate is comprised of 750 Mbps of VoIP and 250 Mbps of standard data traffic then such routing would aggregate 1.5 Gbps of VoIP traffic on the R3 → R4 → R5 links, thereby exceeding our EF class bound of 50%. DS-TE can be used to overcome this problem if, for example, each link is configured with an available aggregate bandwidth pool of 2.5 Gbps,

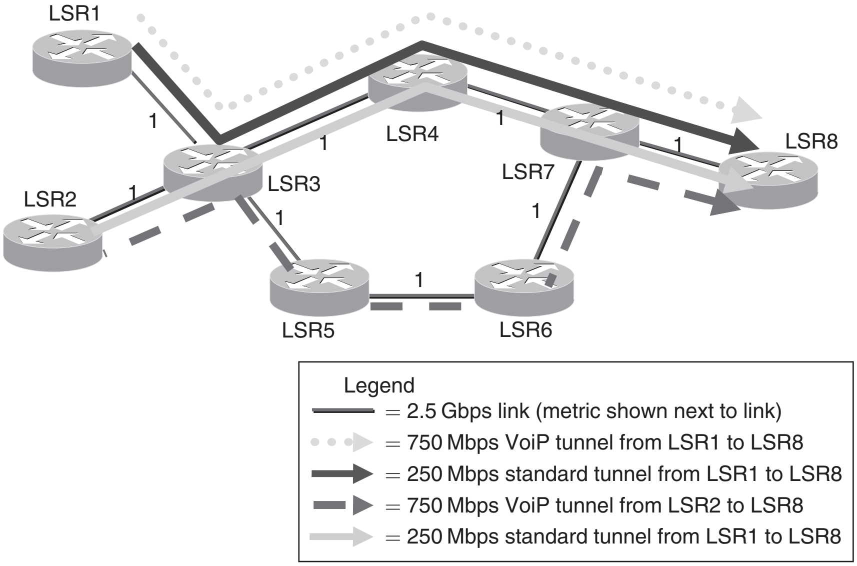

LSR1
LSR2
LSR3
LSR4
LSR5
LSR6
LSR7
LSR8
1
1
1
1
1
1
1
1
Legend
= 2.5 Gbps link (metric shown next to link)
= 750 Mbps VoiP tunnel from LSR1 to LSR8
= 250 Mbps standard tunnel from LSR1 to LSR8
= 750 Mbps VoiP tunnel from LSR2 to LSR8
= 250 Mbps standard tunnel from LSR1 to LSR8

Figure 6.16 DS-TE deployment example 2

and an available VoIP class sub-pool bandwidth of 1.25 Gbps (i.e. 50% of 2.5 Gbps). A VoIP class sub-pool tunnel of 750 Mbps is then configured from R1 to R8, together with a standard class aggregate pool tunnel of 250 Mbps. Similarly, from R2 to R8 a VoIP class sub-pool tunnel of 750 Mbps and a standard class aggregate pool tunnel of 250 Mbps are configured from R2 to R8. The DS-TE constraint-based routing algorithm would then route the VoIP sub-pool tunnels to ensure that the 1.25 Gbps bound is not exceeded on any link, and of the tunnels from R1 and R2 to R8, one VoIP sub-pool tunnel would be routed via the top path (R1/R2 → R3 → R4 → R5 → R8) and the other via the bottom path (R1/R2 → R6 → R7 → R5 → R8).[4] In this particular case, there would be enough available bandwidth for both aggregate pool tunnels to be routed via the top path (R1/R2 → R3 → R4 → R5 → R8), which is the shortest path with available aggregate bandwidth, possibly as shown in Figure 6.16, for example.

Hence, DS-TE allows separate route computation and admission control for different classes of traffic, which enables the distribution of EF and AF class load over all available EF and AF class capacity making optimal use of available capacity. It also provides a tool for constraining the class utilization per link to a specified maximum thus ensuring that the class SLAs can be met. In order to provide these benefits, however, the configured bandwidth for the sub-pools must align to the queuing resources that are available for traffic-engineered traffic.

6.2.3.3 MPLS TE Deployment Models and Considerations

MPLS TE can be deployed either in an ad hoc fashion, with selective tunnels configured tactically to move a subset of traffic away from congested links, or systematically, with all backbone traffic transported in TE tunnels.

6.2.3.3.1 Tactical TE Deployment

MPLS TE can be used tactically in order to offload traffic from congestion hotspots; this is an ad hoc approach, aimed at fixing current problems and as such is generally a short-term reactive operational/engineering process. When used in this way, rather than all traffic being subjected to traffic engineering, TE tunnels are deployed to reroute a subset of the network traffic from a congested part of the network, to a part where there is more capacity. This can be done by explicitly defining the path that a tunnel should take on a head-end router.

Consider Figure 6.17, for example; in this case there are two links of unequal capacity providing the connectivity between two POPs; one 622 Mbps, the other 2.5 Gbps. Using IGP metrics proportional to link capacity, e.g. a link cost of 1 for the 2.5 Gbps links and a link cost of 4 for 622 Mbps link, in normal working case conditions, the bottom path would be the lowest cost path and the top path would remain unused. Hence, even though there is over 3 Gbps of capacity between the POPs, this capacity could not all be used. If, however, two TE tunnels were configured between LSR 1 and LSR 2, one explicitly defined to use the top path and the other the bottom path, then as MPLS TE supports unequal cost load balancing (which normal IGP routing does not), the traffic demand between Router 1 and Router 2 could be balanced over the tunnels in proportion to the bandwidths of those paths, i.e. 1/5 of the total demand using the top path and 4/5 of the total demand on the bottom path.

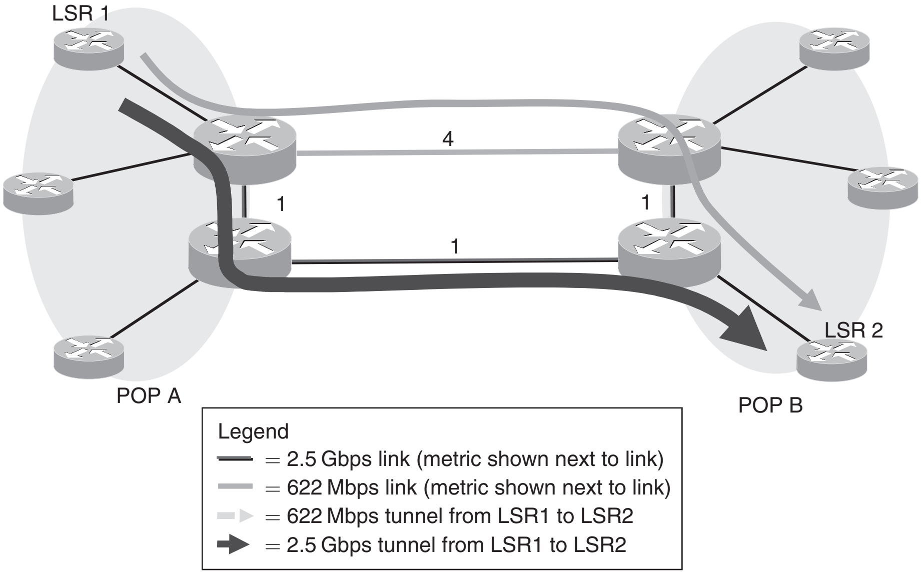

LSR 1
4
1
1
1
LSR 2
POP A
POP B
Legend
= 2.5 Gbps link (metric shown next to link)
= 622 Mbps link (metric shown next to link)
= 622 Mbps tunnel from LSR1 to LSR2
= 2.5 Gbps tunnel from LSR1 to LSR2

Figure 6.17 Tactical TE deployment – enables unequal cost load balancing

6.2.3.3.2 Systematic TE Deployment

With a systematic TE deployment, all traffic is subjected to traffic engineering within the core; this is a long-term proactive engineering/planning process aimed at cost savings. Such a systematic approach requires that a mesh of TE tunnels be configured, hence one of the key considerations for a systematic MPLS TE deployment is tunnel scaling; a router incurs control plane processing overhead for each tunnel that it has some responsibility for, either as head-end, mid-point, or tail-end of that tunnel. The main metrics that are considered with respect to TE tunnel scalability are the number of tunnels per head-end and the number of tunnels traversing a tunnel mid-point. We consider the key scaling characteristics of a number of different systematic MPLS TE deployment models:

- *Outer core mesh.* In considering a full mesh from edge-to-edge across the core (i.e. from distribution router to distribution router), as MPLS TE tunnels are unidirectional, two tunnels are required between each pair of edge routers hence $n * (n - 1)$ tunnels are

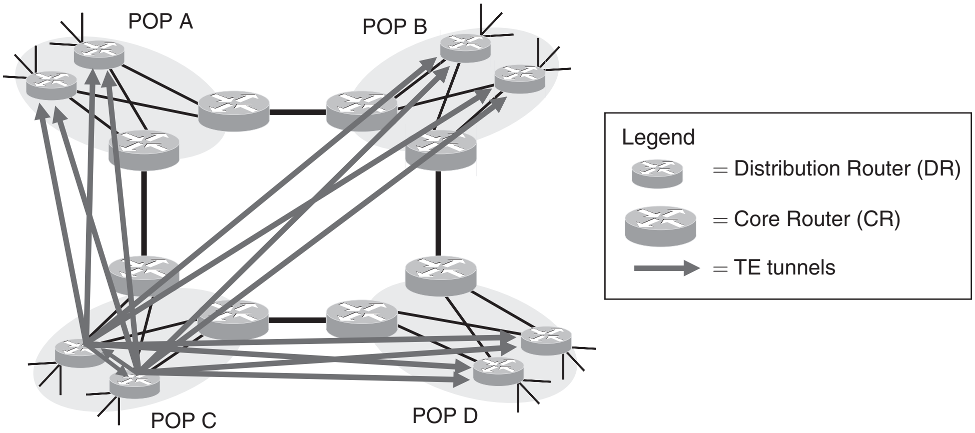


Figure 6.18 Outer core TE mesh

required in total where *n* is the number of edge routers or head-ends. The example in Figure 6.18 shows the tunnels that would be required from the distribution routers within one POP to form a mesh to the distribution routers in other POPs.

If TE is required for *m* classes of traffic each using Diffserv-aware TE then *m* * *n* * *(n − 1)* tunnels would be required.

- *Inner core mesh.* Creating a core mesh of tunnels, i.e. from core routers to core routers, can make tunnel scaling independent of the number of distribution routers (there are normally more distribution routers than core routers), as shown in Figure 6.19, which illustrates the tunnels that would be required from the core routers within one POP to form a mesh to the core routers in other POPs.
- *Regional meshes.* Another way of reducing the number of tunnels required and therefore improving the tunnel scalability is to break the topology up into regions of meshed routers; adjacent tunnel meshes would be connected by routers which are part of both meshes, as shown in Figure 6.20, which shows meshes within each of two regions. Although this reduces the number of tunnels required, it may result in less optimal routing and less optimal use of available capacity.

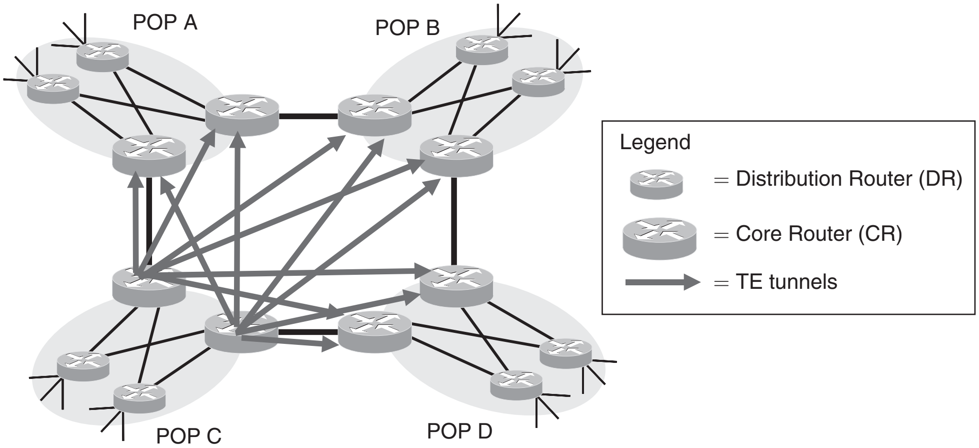


Figure 6.19 Inner core MPLS TE mesh

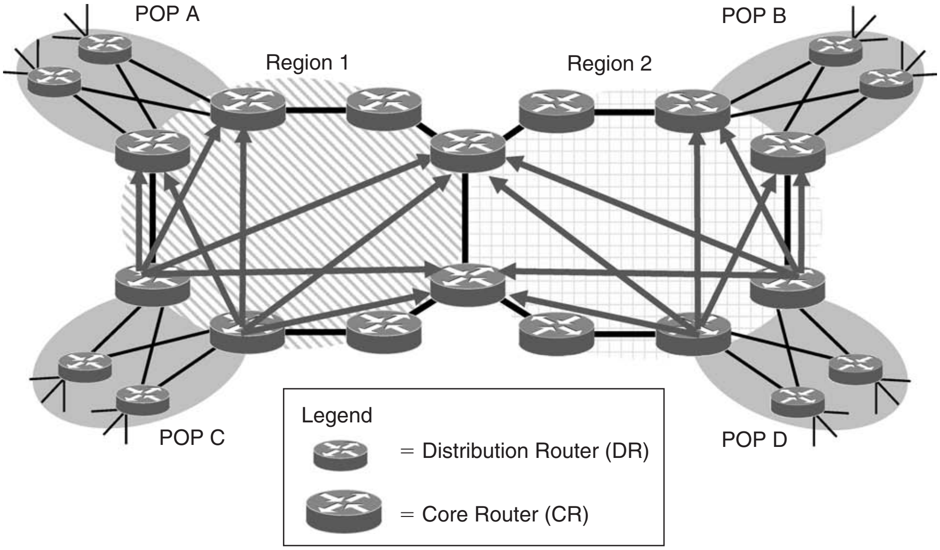


Figure 6.20 Regional MPLS TE meshes

To put these options into context, the largest TE deployments at the time of publication have a full mesh between ∼120 head-ends, which results in $\sim 120^2 = \sim 14\,400$ tunnels in total with a maximum of ∼120 tunnels per head-end and a maximum of ∼1500 tunnels traversing a mid-point.

6.2.3.4 Setting Tunnel Bandwidth

Having decided on a particular MPLS TE deployment model, the next most significant decision is how to set the bandwidth requested for TE tunnels. The bandwidth of tunnels is a logical (i.e. control plane) constraint, rather than a physical constraint, hence if the actual tunnel load exceeds the reserved bandwidth, congestion can occur. Conversely, if a tunnel reservation is greater than the actual tunnel load, more bandwidth may be reserved than is required, which may lead to needless rejection of other tunnels and hence underutilization of the network.

The same principles of over-provisioning discussed in Section 6.1.3 could be applied to traffic engineering deployments. The bandwidth pools on each link should be set taking the required over-provisioning ratios into account for that particular link speed. For example, if Diffserv is not deployed in the core network and an OP of 1.42 is determined to be required to achieve a target P99.9 queuing delay of 2 ms on a 155 Mbps link, then the aggregate TE bandwidth pool should be set to 155/1.42 = 109 Mbps. Each tunnel (which represents a traffic demand across the network) should then be sized based upon the measured average tunnel load (or a percentile thereof, as described for the core traffic demand matrices in Section 6.1.2). This will ensure that the measured average aggregate load on each link will be controlled such that the per-link over-provisioning factor is always met, and hence the target SLAs can be achieved, even when there are potentially multiple tunnels that may traverse the link.

Tunnel resizing can be performed online, by the head-end routers themselves, or by an offline system. When online tunnel resizing is used, algorithms run on the head-end routers to automatically and dynamically resize the tunnels which originate from them, based upon some measure of the traffic load on the tunnel over previous measurement periods. Simple algorithms can lead to inefficiencies, however. Consider, for example, an algorithm that sizes the tunnel based upon the peak of the 5-minute average tunnel loads in the previous interval; when traffic is ramping up during the day, the algorithm needs to take into account the traffic growth during the next interval, or else it will under-provision the tunnel. Consequently,

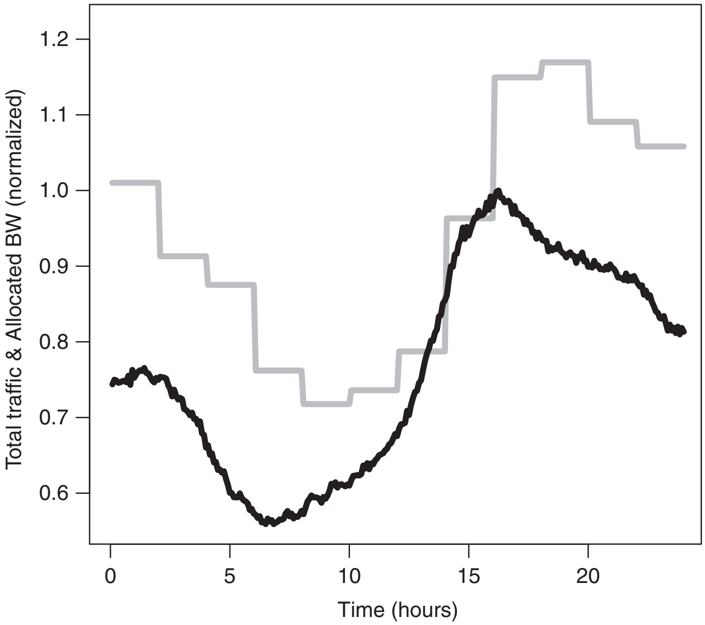


Figure 6.21 Automatic tunnel bandwidth sizing

in the interval following the peak interval of the day, significantly more tunnel bandwidth will be reserved than is necessary, as illustrated by the example in Figure 6.21.

Figure 6.21 plots the total traffic load across all TE tunnels (black line) in a network with a TE tunnel full mesh during a single day. The corresponding sum of the reserved TE tunnel bandwidth is plotted in grey. The tunnel resizing algorithm used in this case resized each tunnel every 2 hours to a multiple of the peak of the 5-minute average load for that tunnel experienced during the preceding 2 hour period. In order to cope with the rapid ramp up in traffic load before the daily peak, a high multiple needed to be used; in this case the multiple was 1.2 times. As a consequence, the reserved tunnel bandwidth is significantly greater than the actual tunnel load during the period after the daily peak load, due to the resizing lag. Hence, tunnel resizing algorithms are most efficient when they rely on a longer history of measurements for tunnel sizing, i.e. day, week, or month.

References[5]

[AMEUR] W. Ben Ameur, N. Michel, E. Gourdin and B. Liau, Routing strategies for IP networks, *Telektronikk*, 2/3, pp. 145–158, 2001

[BONALD] Thomas Bonald, Alexandre Proutiere, James Roberts, Statistical Guarantees for Streaming Flows Using Expedited Forwarding, *INFOCOM 2001*

[BURIOL] L. S. Buriol, M. G. C. Resende, C. C. Ribeiro, and M. Thorup, A memetic algorithm for OSPF routing, in *Proceedings of the 6th INFORMS Telecom*, pp. 187–188, 2002

[CAO] Cao, J., W.S. Cleveland, D. Lin, D.X. Sun, Internet Traffic Tends Toward Poisson and Independent as the Load Increases, in *Nonlinear Estimation and Classification*, New York, Springer-Verlag, 2002

[CHARNY] Anna Charny and Jean-Yves Le Boudec, Delay bounds in a network with aggregate scheduling, in *First International Workshop on Quality of future Internet Services*, Berlin, Germany, 2000

[ERICSSON] M. Ericsson, M. Resende, and P. Pardalos, A genetic algorithm for the weight setting problem in OSPF routing, *J. Combinatorial Optimization*, volume 6, no. 3, pp. 299–333, 2002

[FORTZ1] B. Fortz, J. Rexford, and M. Thorup, Traffic Engineering With Traditional IP Routing Protocols, *IEEE Communications Magazine*, October 2002

[FORTZ2] Bernard Fortz, Mikkel Thorup, Internet traffic engineering by optimizing OSPF weights, *Proc. IEEE INFOCOM, 2000*, pp. 519–528, March 2000

[FRALEIGH] Chuck Fraleigh, Fouad Tobagi, Christophe Diot, Provisioning IP Backbone Networks to Support Latency Sensitive Traffic, *Proc. IEEE INFOCOM 2003*, April 2003

[GOUS] Alan Gous, Arash Afrakhteh, Thomas Telkamp, Traffic Engineering through Automated Optimization of Routing Metrics, presented at Terena 2004 conference, Rhodes, June 2004. Available at:

http://tnc2004.terena.nl/programme/presentations/show.php?pres_ id = 99

[HALABI] Sam Halabi, *Internet Routing Architectures*, Cisco Press, 2000

[IPFIX] B. Claise, Ed., Specification of the IPFIX Protocol for the Exchange of IP Traffic Flow Information Protocol Specification, IETF draft draft-ietf-ipfix-protocol, November 2006 [work in progress]

[LORENZ] D. Lorenz, A. Ordi, D. Raz, and Y. Shavitt, How good can IP routing be?, *DIMACS Technical, Report 2001-17*, May 2001

[MAGHBOULEH] Arman Maghbouleh, Metric-Based Traffic Engineering: Panacea or Snake Oil? A Real-World Study, *Arman Maghbouleh, Cariden*, NANOG 27, February 2003

[MEDINA] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot, Traffic matrix estimation: Existing techniques and new directions, in *ACM SIGCOMM* (Pittsburg, USA), August 2002

[PAXON] V. Paxson and S. Floyd, Wide-area traffic: The failure of Poisson modeling, *IEEE/ACM Transactions on Networking*, vol. 3, no. 3, pp. 226–244, 1994

[RFC1142] D. Oran, Ed., OSI IS-IS Intra-domain Routing Protocol, RFC 1142, February 1999 [republication of ISO DP 10589]

[RFC2205] R. Braden, Ed., Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification, *RFC 2205*, September 1997

[RFC2328] J. Moy, OSPF version 2, *RFC 2328*, April 1998

[RFC3036] L. Andersson et al., LDP Specification, *RFC 3036*, January 2001

[RFC3209] D. Awduche et al., RSVP-TE: Extensions to RSVP for LSP tunnels, *RFC 3209*, Dec. 2001; http://www.rfc-editor.org/rfc/rfc3209.txt

[RFC3630] D. Katz, K. Kompella, D. Yeung, Traffic Engineering (TE) Extensions to OSPF Version 2, *RFC 3630*, September 2003

[RFC3784] H. Smit, T. Li, Intermediate System to Intermediate System (IS-IS) Extensions for Traffic Engineering (TE), *RFC 3784*, June 2004

[RFC3785] Le Faucheur et al., Use of IGP Metric as a second TE Metric, *RFC 3785*, May 2004

[RFC3906] N. Shen, H. Smit, Calculating Interior Gateway Protocol (IGP) Routes Over Traffic Engineering Tunnels, *RFC 3906*, October 2004

[RFC4124] F. Le Faucheur, Ed., Protocol extensions for support of Diff-Serv-aware MPLS Traffic Engineering, *RFC 4124*, June 2005

[RFC4125] F. Le Faucheur, W. Lai, Maximum Allocation Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering, *RFC 4125*, June 2005

[RFC4127] F. Le Faucheur, Ed., Russian Dolls Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering, *RFC 4127*, June 2005

[RFC4271] Y. Rekhter, Ed., T. Li, Ed., S. Hares, Ed., A Border Gateway Protocol 4 (BGP-4), *RFC 4271*, January 2006

[RFC4364] E. Rosen, Y. Rekhter, BGP/MPLS IP Virtual Private Networks (VPNs), *RFC 4364*, February 2006

[SAHINOGLU] Z. Sahinoglu, and S. Tekinay, On Multimedia Networks: Self-Similar Traffic and Network Performance, *IEEE Communications Magazine*, pp. 48–52, January 1999

[TEBALDI] C. Tebaldi and M. West, Bayesian inference on network traffic using link count data, *J. Amer. Statist. Assoc.*, vol. 93, no. 442, pp. 557–576, 1998

[TELKAMP1] Thomas Telkamp, Best Practices for Determining the Traffic Matrix in IP Networks V 2.0, NANOG 35 Los Angeles, October 2005. Available online at: http://www.nanog.org/mtg-0510/telkamp.html

[TELKAMP2] Thomas Telkamp, Traffic Characteristics and Network Planning, NANOG 26, October 2002. Available at: http://www.nanog.org/mtg-0210/telkamp.html

[VARDI] Y. Vardi, Network tomography: estimating source-destination traffic intensities from link data, *J. Am. Statist. Assoc.*, vol. 91, pp. 365–377, 1996

[ZHANG1] Y. Zhang, M. Roughan, N. Duffeld, and A. Greenberg, Fast accurate computation of large-scale IP traffic matrices from link loads, in *ACM SIGMETRICS* (San Diego, California), pp. 206–217, June 2003

[ZHANG2] Z.-L. Zhang, V. Ribeiro, S.Moon, and C. Diot, Small-Time Scaling behaviors of internet backbone traffic: An Empirical Study. In *IEEE Infocom*, San Francisco, Mar. 2003

Notes

1. The distribution routers in the generalized network reference model we use in this book will normally be provider edge (PE) routers in the context of an MPLS VPN deployment.
2. Defined by the solution to the maximum multicommodity flow problem, where the total flow summed over all commodities is to be maximized.
3. Route pinning is the ability to explicitly define the exact path that a particular traffic flow may take through the network.
4. A propagation-delay constraint can also be specified for the subpool tunnels to ensure that the chosen path exhibits a propagation delay smaller or equal to the specified value [RFC3785].
5. The nature of the networking industry and community means that some of the sources referred to in this book exist only on the World Wide Web. All universal resource locators (URLs) have been checked and were correct at the time of going to press, but their longevity cannot be guaranteed.