1

Foundations

1.1 HISTORICAL PERSPECTIVE

In only a few years, Multi-Protocol Label Switching (MPLS) has evolved from an exotic technology to a mainstream tool used by service providers to create revenue-generating services. There is rapid deployment of MPLS-enabled services and active development of new mechanisms and applications for MPLS in the standards bodies. This book aims to describe the fundamental mechanisms used by MPLS and the main service types that MPLS enables, such as Virtual Private Networks (VPNs). We include descriptions of new applications of MPLS that are currently under development.

The history of MPLS and its precursors is described in [Davie Rekhter] and [Doyle Kolon]. The first Internet Engineering Task Force (IETF) MPLS Working Group Meeting took place in April 1997. That working group still exists, and MPLS has grown to the extent that it underpins much of the activity of several other working groups in the IETF, such as Layer 3 VPN (l3vpn), Layer 2 VPN (l2vpn), Pseudo Wire Emulation Edge to Edge (pwe3) and Common Control and Measurement Plane (ccamp). Part of the original MPLS problem statement [MPLS97] from the first MPLS working group meeting is shown below. It contains four items that the group aimed to address through the development of MPLS. It is interesting to examine these to see which items are still relevant today:

1. *Scalability of network layer routing*. Using labels as a means to aggregate forwarding information, while working in the presence of routing hierarchies.

MPLS-Enabled Applications: Emerging Developments and New Technologies Second Edition
Ina Minei and Julian Lucek © 2008 John Wiley & Sons, Ltd

Layer 3 VPNs have proved to be a good example of aggregation of forwarding information. As described in Chapter 7 of this book, edge routers need to contain routing information pertaining to each VPN that they service, but the core routers do not. Thus, assuming that any edge router services only a subset of the VPNs pertaining to the network, no router in the network needs to hold the entire set of routes present in the network.

2. *Greater flexibility in delivering routing services*. Using labels to identify particular traffic which are to receive special services, e.g. QoS. Using labels to provide forwarding along an explicit path different from the one constructed by destination-based forwarding.

MPLS has the ability to identify particular traffic flows which must receive special services such as Quality-of-Service (QoS). It also has traffic engineering properties that allow it to provide forwarding along a particular explicit path. These two properties are combined in DiffServ Aware Traffic Engineering, which is described in more detail in Chapter 4 of this book.

3. *Increased performance*. Using the label-swapping paradigm to optimize network performance.

Because modern routers perform packet forwarding in hardware, the forwarding rates for IP and MPLS packets are similar. However, 'optimizing network performance' implies a wider context than simply the performance of individual nodes. Certainly MPLS has helped in this wider context, e.g. through the use of traffic engineering to avoid congestion and the use of fast reroute to reduce the interruption to traffic when a link in the network fails.

4. *Simplify integration of routers with cell switching based technologies*: a) making cell switches behave as peers to routers (thus reducing the number of routing peers that a router has to maintain), b) by making information about physical topology available to Network Layer routing procedures, and c) by employing common addressing, routing, and management procedures.

When this item in the problem statement was written, many networks had a core of asynchronous transfer mode (ATM) switches surrounded by routers. The routers were typically fully meshed with ATM connections. This overlay model was proving difficult to scale because the number of routing adjacencies required grew as the square of the number of routers involved; hence there was a requirement to make the ATM switches act as peers to the routers. It is interesting to note that the situation has now been turned inside out: now many networks have an MPLS-based core, and service providers are migrating ATM services to this core network by

interconnecting ATM switches with Layer 2 connections over the MPLS core! This has the problem that the number of adjacencies between ATM switches grows as the square of the number of ATM switches involved. Hence, currently there is work on making ATM switches behave as peers to routers [MPLS ALLI]. This is to avoid having a full mesh of adjacencies between ATM switches rather than to avoid having a full mesh of adjacencies between routers, as stated in the problem statement. The concept expressed in the problem statement of using MPLS as a control plane for multiple technologies has manifested itself in Generalized MPLS (GMPLS). In GMPLS, a common control plane covers a wide range of network devices, such as routers, ATM switches, SONET/SDH equipment and optical cross-connects [RFC3945].

In summary, much of the original problem statement is still relevant today. Many of the mechanisms of MPLS described in Part 1 of this book were developed to address the items listed above, to the benefit of the MPLS applications discussed in Part 2 of this book.

1.2 CURRENT TRENDS

At the time of writing this book, the most widely deployed customer-visible MPLS service is the Layer 3 VPN (also known as an IP VPN or 2547bis VPN, after the IETF document describing them). MPLS is also used in some networks as an infrastructure tool to provide traffic engineering and fast-reroute capabilities. Another rapidly growing application is point-to-point Layer 2 transport, either as means of carrying a customer's Ethernet traffic across the wide area or as a component of ATM or Frame Relay Service emulation. Finally, Virtual Private LAN Service (VPLS) offerings, in which the service provider gives the impression to the customer that their sites are attached to the same Local Area Network (LAN), are also becoming available.

Many service providers are investigating the possibility of using an MPLS-based network to provide a common platform for a wide range of services that are currently typically delivered over multiple distinct networks. Such a multiservice network might carry Public Switched Telephone Network (PSTN) traffic, public Internet and private IP data services, Layer 2 ATM and Frame Relay services, Broadcast TV and TDM traffic. This offers capital and operational cost savings to the network operators by allowing them to operate a single network rather than a separate network for each service type. A key aim of this book is to show how MPLS can provide the necessary mechanisms for this network convergence, e.g. through the use of DiffServ Aware Traffic Engineering (TE), which allows the MPLS network to provide connection-orientated characteristics to particular traffic flows.

1.3 MPLS MECHANISMS

This section gives an overview of the mechanisms underpinning MPLS. Readers who are familiar with these may wish to skip this section.

A fundamental property of an MPLS network is that it can be used to tunnel multiple traffic types through the core of the network. Tunneling is a powerful tool because only the routers at the ingress and the egress of the tunnel need to understand the 'context' of the underlying traffic carried over the tunnel (e.g. the protocol that the traffic belongs to and the reachability information required to route and forward it in its native form). This detail is hidden from routers in the core of the network. As a consequence, core devices only need to carry sufficient state to enable them to switch MPLS-encapsulated packets without regard to their underlying content. Besides these aggregation properties, which apply to tunnels in general, MPLS tunnels have the following particular properties:

1. Traffic can be explicitly routed, depending on which signaling protocol is used.
2. Recursion is provided for; hence tunnels can exist within tunnels.
3. There is protection against data spoofing, as the only place where data can be injected into an MPLS tunnel is at the head end of that tunnel. In contrast, data can be injected into an IP tunnel from any source that has connectivity to the network that carries the tunnel.
4. The encapsulation overhead is relatively low (4 bytes per MPLS header).

An MPLS network consists of edge devices known as Label Edge Routers (LERs) or Provider Edge (PE) routers and core routers known as Label Switching Routers (LSRs) or Provider (P) routers. A mesh of unidirectional tunnels, known as Label Switched Paths (LSPs) is built between the LERs in order that a packet entering the network at the ingress LER can be transported to the appropriate egress LER. When packets enter a network, the ingress router determines which Forwarding Equivalence Class (FEC) the packets belong to. Packets that are to be forwarded to the same egress point in the network along the same path and with the same forwarding treatment along that path are said to belong to the same FEC. Packets belonging to the same FEC are forwarded with the same MPLS label. In a simple case, packets whose destination addresses correspond to the same Border Gateway Protocol (BGP) next-hop are regarded by the ingress router as belonging to the same FEC. In other cases, there may be a more granular assignment of packets to FECs. For example, in DiffServ Aware TE, each egress point in the network may have multiple FECs, each belonging to a different traffic class.

It is the role of the ingress LER to determine the appropriate egress LER and LSP to that egress LER associated with the FEC. MPLS has the

property that multiple traffic types can be multiplexed on to a single LSP. Therefore, if desired by the network operator, a single LSP can be used to carry all the traffic (e.g. L3VPN, public IP and Layer 2) between a particular ingress LER and a particular egress LER. Transit routers along the path of the LSP make their forwarding decision on the basis of a fixed-format MPLS header, and hence do not need to store 'routes' (L3VPN routes, external IP routes, Layer 2 forwarding information) pertaining to the underlying tunneled packets. This is an important scaling property, as otherwise each of the core routers would have to carry routing information equivalent to the sum of the routing information carried by all the edge routers in the network.

The following sections describe the fundamental forwarding plane and control plane mechanisms underpinning MPLS.

1.3.1 Forwarding plane mechanisms

Data carried over an MPLS-capable network has one or more MPLS headers applied in order to transport it across the network. The MPLS header structure is shown in Figure 1.1. It contains the following fields:

1. *A 20-bit label value*. MPLS packets are forwarded on the basis of this field. This value is used as an index into the MPLS forwarding table.
2. *EXP field (3 bits)*. These bits are known as the experimental bits. In practice, they are used to convey the Class of Service to be applied to the packet. For example, LSRs and LERs can use these bits to determine the queue into which the packet should be placed. Note that in some cases, as described later in this chapter, the MPLS label value also determines the queuing behavior applied to the packet.
3. *Bottom of stack bit (S-bit)*. As described later in this chapter, MPLS headers can be stacked. The S-bit is set on the header of the MPLS packet at the bottom of the stack.
4. *Time-to-live (TTL) field*. This is used to avoid forwarding loops and can also be used for path-tracing. The value is decremented at each hop and the packet is discarded should the value reach zero.

Packets arriving into the network have one or more MPLS headers applied by the ingress LER. The ingress LER identifies the egress LER to which the packet must be sent and the corresponding LSP. The label

Figure 1.1 MPLS header structure

value used corresponds to the LSP on to which the packet is placed. The next router performs a lookup of that label and determines the output label that must be used for the next leg of the LSP. The lookup operation on a P router involves reading the incoming label; this yields a new label value to use and the output interface(s) on which the packet should be forwarded. In this way, through this label-swapping paradigm, the packet is conveyed along the LSP from the ingress to the egress LER.

In some simple cases, the use of a single MPLS label is sufficient, e.g. when transporting public IP traffic across a network. In this case, once the packet arrives at the egress LER, the LER performs a normal IP lookup in order to determine which egress link to use. Usually a scheme called Penultimate Hop Popping (PHP) is used. In this scheme, the LSR before the egress LER (i.e. the penultimate router along the LSP) pops the MPLS label and forwards it to the egress LER as an IP packet. This simplifies the processing required at the egress node, as otherwise it would be necessary to pop the label and perform an IP lookup at the egress node. It is not mandatory for the egress router to request PHP behavior, but is the default behavior of most implementations.

In other cases, a single MPLS header is insufficient. This is because the LERs in a particular network may be involved in multiple services – Layer 3 VPN, Layer 2 VPN, VPLS – rather than just the public IP. In this case, the egress LER needs to know which service and which instance of that service (i.e. which customer) the packet belongs to. This is achieved by having an additional MPLS header, which is applied by the ingress LER, corresponding to the service and service instance that the packet must be directed to by the egress LER once the packet has crossed the network. This is illustrated in Figure 1.2.

Let us see how an MPLS packet with two headers is transported between the ingress and egress LERs. The inner header with label Y denotes the service and service instance, and the outer header, often called the 'transport' header, is the one required to transport the packet from the ingress LER, PE1, to the correct egress LER, PE2. For example, a particular LER may be running several Layer 3 VPN, VPLS and Layer 2 VPN instances. Label Y tells the egress LER that the packet in question corresponds to the Layer 3 VPN service being provided to Company A, rather than any of the other Layer 3 VPN instances or the VPLS or Layer

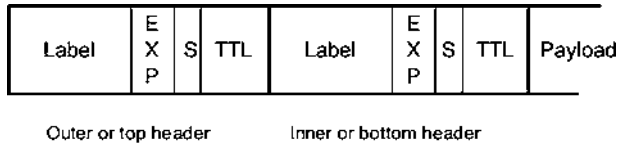

Figure 1.2 MPLS header stack

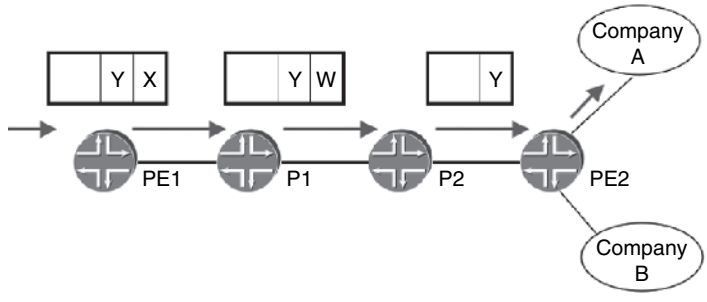

Figure 1.3 Forwarding a packet having two MPLS headers

2 VPN instances. The ability to stack headers in this way gives MPLS key multiplexing and hierarchical properties, allowing a single LSP between a particular ingress and egress point to carry all traffic between those points. As Figure 1.3 shows, the packet leaves the ingress LER, PE1, with an inner label value of Y and an outer label value of X. Routers P1 and P2 perform a lookup based on the outer transport label and do not need to read or take any action based on the inner label. P1 swaps outer label X with outer label W. If PHP is in use, which is typically the case, router P2 pops the outer header, and sends the remainder of the packet to PE2. Thus, when the packet arrives at PE2, the outermost (and only) label is the original inner label, Y, which PE2 uses to identify the packet as belonging to the Layer 3 VPN instance pertaining to Company A.

How does the ingress LER know the label value(s) to use? The transport label is learnt through either the RSVP or LDP signaling protocols, which are described in more detail later on in this chapter. The inner label in the case of most services is learnt via BGP (e.g. Layer 3 VPNs, BGP-signaled Layer 2 VPNs). However, there are also cases where LDP is used, e.g. LDP-signaled Layer 2 transport circuits.

1.3.1.1 MPLS support of DiffServ

DiffServ was developed as a solution to provide Quality-of-Service (QoS). It does so by dividing traffic into a small number of classes and allocating network resources on a per-class basis. To avoid the need for a signaling protocol, the class is marked directly within the packet header. The DiffServ solution was targeted at IP networks so the marking is in the 6-bit DiffServ Code Point (DSCP) field in the IP header. The DSCP determines the QoS behavior of a packet at a particular node in the network. This is called the per-hop behavior (PHB) and is expressed in terms of the scheduling and drop preference that a packet experiences. From an

implementation point of view, the PHB translates to the packet queue used for forwarding, the drop probability in case the queue exceeds a certain limit, the resources (buffers and bandwidth) allocated to each queue and the frequency at which a queue is serviced.

The first challenge with supporting DiffServ in an MPLS network is that LSRs make their forwarding decisions based on the MPLS header alone, so the per-hop behavior (PHB) needs to be inferred from it. The IETF solved this problem by assigning the three experimental (EXP) bits in the MPLS header to carry DiffServ information in MPLS.

This solution solves the initial problem of conveying the desired PHB in the MPLS header, while introducing a new one: how does one map DSCP values expressed in a 6-bit field that can encode up to 64 values into a 3-bit EXP field that can carry at most eight distinct values? There are two solutions to this problem, discussed separately below.

The first solution applies to networks that support less than eight PHBs. Here, the mapping is straightforward: a particular DSCP is equivalent to a particular EXP combination and maps to a particular PHB (scheduling and drop priority). During forwarding, the label determines where to forward the packet and the EXP bits determine the PHB. The EXP bits are not a property that is signaled when the label-switched path (LSP) is established; the mapping of EXP to PHB is configured on each node in the network. The EXP bits can be set according to the DSCP bits of the IP packets carried in the LSP, or they can be set by the network operator. LSPs for which the PHB is inferred from the EXP bits are called E-LSPs (where E stands for 'EXP-inferred'). E-LSPs can carry packets with up to eight distinct per-hop behaviors in a single LSP.

The second solution applies to networks that support more than eight PHBs. Here, the EXP bits alone cannot carry all the necessary information to distinguish between PHBs. The only other field in the MPLS header that can be used for this purpose is the label itself. During forwarding, the label determines where to forward the packet and what scheduling behavior to grant it, and the EXP bits convey information regarding the drop priority assigned to a packet. Thus, the PHB is determined from both the label and the EXP bits. Because the label is implicitly tied to a per-hop behavior, this information needs to be conveyed when the LSP is signaled. LSPs that use the label to convey information about the desired PHB are called L-LSPs (where L stands for 'label-inferred'). L-LSPs can carry packets from a single PHB or from several PHBs that have the same scheduling regimen but differ in their drop priorities (e.g. the set of classes AF*xy* where *x* is constant are treated the same from the scheduling point of view but differ in their drop priority according to the value of *y*). Table 1.1 summarizes the differences between E-LSPs and L-LSPs.

Table 1.1 Comparison of E-LSPs and L-LSPs

E-LSP	L-LSP
PHB is determined by the EXP bits	PHB is determined by the label or by the label and EXP bits together
Can carry traffic with up to 8 distinct PHBs in a single LSP	Can carry a single PHB per LSP or several PHBs with the same scheduling regimen and different drop priorities
User conservative label and maintains state, because the label is used only for conveying path information	Uses more labels and keeps more state, because the label conveys information about both the path and the scheduling behavior
No signaling is required to convey the PHB information	The PHB information needs to be signaled when the LSP is established
Up to 8 PHBs can be supported in the network when only E-LSPs are used. E-LSPs can be used in conjunction with L-LSPs when more PHBs are required	Any number of PHBs can be supported in the network

1.3.2 Control plane mechanisms

So far we have seen how MPLS uses labels for forwarding, but how are the bindings between labels and FECs distributed throughout the network? Since manual configuration is not an option, there clearly is a need for a protocol to disseminate this information. From a practical point of view, there are two options: (a) invent a new protocol for distributing label bindings or (b) extend an existing protocol to carry labels in addition to routing information. The question of whether to invent a new protocol or extend an existing one is a popular one in the MPLS world, and we will discuss it in detail in later chapters. At this point, suffice it to say that when the question arises, the result is usually that both approaches are followed.

Regarding the distribution of label bindings, the engineering community invented a new protocol (LDP, or Label Distribution Protocol) and extended two existing protocols (RSVP, or Resource Reservation Protocol, and BGP, or Border Gateway Protocol). The packet formats and basic operation of these protocols are explained in detail in many introductory texts [Doyle Kolon, Osborne Simha]. Instead of repeating this information here, let us instead examine the properties of the different protocols, and see the benefits and limitations of each of them.

1.3.2.1 LDP

LDP [RFC5036] is the result of the MPLS Working Group [MPLS WG] in the IETF. Unlike RSVP or BGP, which existed well before MPLS and were extended to do label distribution, LDP was specifically designed to distribute labels in the network. Since the goal of LDP is label distribution, LDP does not attempt to perform any routing functions and relies on an Interior Gateway Protocol (IGP) for all routing-related decisions. The original LDP specification was defined for setting up LSPs for FECs representing an IPv4 or IPv6 address. This is the functionality described in this section. The extensions of LDP used for pseudo-wire and VPLS signaling will be discussed in the appropriate chapters.

LDP was designed with extensibility in mind. All the information exchanged in LDP is encoded as TLVs (type–length–value triplets). The type and length are at the start of the encoding, and their length is known in advance. The type identifies which information is exchanged and determines how the rest of the encoding is to be understood. The value is the actual information exchanged and the length is the length of the value field. TLVs make it easy to: (a) add new capabilities by adding a new type and (b) skip unknown objects by ignoring the amount of data specified in the length field. Over the years, many new capabilities were added to the protocol thanks to this built-in extensibility.

LDP operation is driven by message exchanges between peers. Potential peers, also known as neighbors, that are directly connected to each other over a point-to-point or LAN interface are automatically discovered via hello messages multicast to a well-known UDP port. The protocol also allows for discovery of remote peers using *targeted* hello messages. In that case, unicast UDP hello messages are sent to the remote neighbor address and may travel through multiple hops to reach the peer.[1] Either way, once a potential peer is discovered, a TCP connection is established to it and an LDP session is set up. If a pair of peers are directly connected over more than one interface, although LDP hellos are exchanged on all those interfaces, there is only one LDP session between them. At session initialization time, the peers exchange information regarding the features and mode of operation they support. After session setup, the peers exchange information regarding the binding between labels and FECs over the TCP connection. The use of TCP ensures reliable delivery of the information and allows for incremental updates, rather than periodic refreshes. LDP uses the regular receipt of protocol messages to monitor the health of the session. In the absence of any new information that needs to be communicated between the peers, keepalive messages are sent.

[1] One case in which targeted hello messages are used is the case of LDP over RSVP tunneling, which is discussed in Section 1.3.2.3 of this chapter.

The association between an FEC and a label is advertised via label messages: label mapping messages for advertising new labels, label withdraw messages for withdrawing previously advertised labels, etc. The fundamental LDP rule states that LSR A that receives a mapping for label L for FEC F from its LDP peer LSR B will use label L for forwarding if and only if B is on the IGP shortest path for destination F from A's point of view. This means that LSPs set up via LDP always follow the IGP shortest path and that LDP uses the IGP to avoid loops.

Relationship between LDP and the IGP

The fact that LDP relies on the IGP for the routing function has several implications:

1. LDP-established LSPs always follow the IGP shortest path. The LSP path shifts in the network when the IGP path changes, rather than being nailed down to a predefined path.
2. The scope of LDP-established LSPs is limited to the scope of the IGP. Thus, LDP LSPs cannot traverse autonomous system (AS) boundaries. The need for Inter-AS LSPs, as well as the solution proposed by the IETF for establishing them, is explained in the Interdomain Traffic Engineering chapter of this book (Chapter 5).
3. During reconvergence, traffic may be blackholed or looped. The existence of loops and the possibility of blackhole traffic is a fact of life for the IGPs during reconvergence. The same properties are inherited by LDP, by virtue of it relying on the IGP for routing decisions. We will discuss how such loops are created and what their impact is in the Protection and Restoration chapter of this book (Chapter 3).
4. The IGP convergence time poses a lower bound on the LDP convergence time. Assuming that the IGP implements smart fast-convergence mechanisms the traffic loss is in the range of 1–2 seconds, orders of magnitude larger than RSVP's fast-reroute time. The IETF is currently working on adding fast-reroute capabilities to LDP. This is discussed in more detail in the Protection and Restoration chapter of this book (Chapter 3).
5. Loss of synchronization between the IGP and LDP can result in traffic loss. As always, for situations where two protocols must operate in tandem, there is a potential for race conditions.

Let us take a closer look at a race condition caused by the loss of synchronization between LDP and the IGP. In the diamond-shaped topology in Figure 1.4, LSR A is advertising a binding for its loopback FEC A. To start with, all links have the same metric, and the link C–D does not exist in the topology. From D's point of view, the LSP for FEC A follows the path D–B–A. At a later time the link C–D is added to the

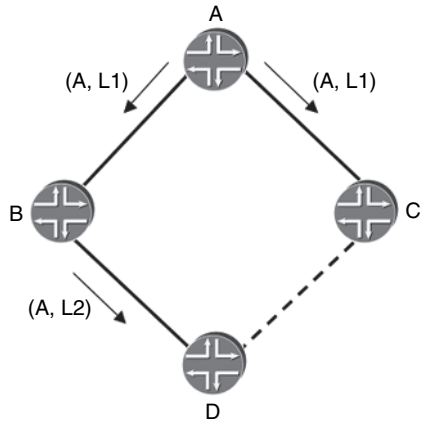

Figure 1.4 Race condition between the IGP and the LGP

topology with a metric that is better than the metric of link B–D, causing the IGP shortest path from D's point of view to be D–C–A. Assume that the IGP reacts faster than LDP. As soon as D finds out about the routing change, it stops using the binding it received from B, thus breaking the LSP. The LSP stays down until a binding for FEC A is received on the LDP session C–D. This may take a while, depending on how fast the session establishment takes place. The situation described here is particularly unattractive, since an alternate path exists in the topology and could have been used until the LDP session comes up on the link C–D.

The above example shows a loss of synchronization caused by the fact that the LDP session on the new link comes up after the IGP session. This is not the only way in which loss of synchronization can occur: forgetting to enable LDP on the new interface, mis-configuring the LDP session authentication, setting up firewall filters that block LDP traffic, or any other event that would cause the IGP to take into account a link but would cause LDP not to use the link, has the same effect.

One solution to this problem is to tie (through configuration) the IGP metric for a particular link to the existence of an LDP session on the link [LDP-IGP-SYNC]. When the LDP session is down, the IGP metric advertised for the link is very high. Therefore, if an alternate path is available, the LDP labels on that path can be used. This is discussed in more detail in the MPLS Management chapter of this book (Chapter 13).

Let us now suppose that the link between C and D is operational but undergoes a flap. That is to say, the link goes down and comes up again a few seconds later. Although the technique described in [LDP-IGP-SYNC] prevents blackholing of traffic while the session between C and D re-establishes and labels are exchanged, the traffic could be

following a suboptimal path through the network for several seconds during this time. An additional technique called 'LDP Session Protection' is supported by some LDP implementations to avoid this problem. This works as follows. While the link between C and D is up, they exchange regular hellos in the normal way. When LDP Session Protection is in use, in addition, C and D also exchange *targeted* hellos. Although there are two types of hello message being exchanged, there is only one LDP session between C and D. If the link between C and D fails, the regular hellos can no longer propagate, but as long as there is still IP connectivity between C and D (via A and B in the example), the targeted hellos can continue to travel between C and D so the LDP session stays up. This means that when the link between C and D subsequently comes up, the session does not need to be re-established or label bindings exchanged. Once regular LDP hello messages have been exchanged over the link, the link can be used for forwarding once more.

So far we have seen the implications of having LDP rely on the IGP for the routing function. Next, let us take a look at the choice of label distribution and retention modes made by common LDP implementations.

Label retention and label distribution modes

Label retention mode – which labels to keep? The LDP specification allows the use of both liberal and conservative label retention modes. Conservative retention means keeping only those labels which are used for forwarding, and discarding the rest. This policy makes sense for devices where the label space is a precious resource that must be carefully managed (such as ATM switches). The savings in the label usage come at a cost. Since the 'uninteresting' labels are discarded, they must be requested again if they become 'interesting' at a later point (e.g. due to a change in routing). Until the requested label arrives, traffic is lost. This undesirable property, coupled with the fact that label space is not a concern in modern routers means that most implementations today use liberal retention.

Label distribution mode – who assigns the labels? The key function of LDP is to distribute bindings between labels and FECs. The goal is to build a forwarding table containing a mapping between an incoming label and an outgoing label. Traffic arriving at the LSR labeled with the incoming label is forwarded labeled with the outgoing label. When building the forwarding table, the question is whether to use the locally picked label as the incoming or the outgoing label. The MPLS architecture [RFC3031] uses downstream label assignment, which means that the router expects to receive the traffic with the label that it picked locally. For example, if LSR A receives label L1 for FEC F and advertises label L2 for it, then it expects traffic destined for FEC F to come labeled with label L2. When forwarding traffic for FEC F, LSR A labels the traffic with label L1. The

traffic flows in the opposite direction from the distribution of labels. The method is called downstream because the label that is assigned to the traffic at point P in the network was actually picked by a router who is one hop further down in the direction of the traffic flow (downstream) from P.

The next question is: should labels be advertised only to those asking for them (on-demand label distribution) or to everyone (unsolicited label distribution)? We have already seen that on-demand label distribution has the undesirable property that traffic is blackholed until the request for the label is satisfied. For this reason, most implementations use the unsolicited label distribution mode. Since LDP uses downstream label allocation, the label distribution mode is usually referred to as downstream unsolicited.

Liberal retention, coupled with unsolicited label advertisements, ensures that labels received from peers are readily available. This is important for handling routing changes in a seamless fashion. To better understand this, let us look at LSR A, which receives two unsolicited label advertisements for FEC F: one with label L1 from peer B and one with label L2 from peer C. LSR A keeps both labels, since it is doing liberal retention. Assuming that the IGP route for FEC F points to peer B, LSR A installs label L1 in its forwarding table. If at some later point the IGP route changes and starts pointing at peer C, all that LSR A has to do is change its forwarding table to use label L2.

Control over the LSP setup

The sole purpose of distributing bindings between labels and FECs is to establish label-switched paths in the network. So far we have discussed a lot of interesting properties of LDP but have not yet answered two key questions: (a) which FEC to advertise a binding for and (b) when to advertise this binding.

The choice of FECs is derived from the LSPs that must be set up in the network. It is independent of the LDP protocol and therefore the LDP specification is silent on this topic. All vendors allow control over the choice of FECs through configuration, but the behavior in the absence of a user-defined configuration is different for different vendors. Some advertise a binding for every prefix in their routing table, while others only advertise a binding for the FEC corresponding to the LSR's loopback address. The outcome in terms of the numbers of LSPs that are set up and of the destinations reachable via these LSPs is quite different. There is no right or wrong decision here, as different implementations may have different constraints. However, from a network operations point of view, it is a bad idea to allow LDP to advertise bindings for FECs that will not be used for forwarding. The extra binding and LSP information uses up resources in the network and makes troubleshooting extremely difficult.

The choice of FEC determines which LSPs are set up. The decision when to advertise the label binding determines who has control over the LSP setup. The LDP specification allows two modes of operation: ordered control and independent control. Since not all vendors implement the same mode, let us take a closer look at the two options and their properties, by reference to Figure 1.5. For the purposes of this discussion, assume that link if5 does not exist. This link will be used for a later discussion in this section.

Ordered control. Under ordered control, egress LSR PE1 initiates the LSP setup by assigning label L1 to the FEC corresponding to its loopback address PE1 and advertising this mapping to its peer A. Upon receipt of the label mapping, A evaluates whether PE1 is on the IGP shortest path for that FEC. Since the check is successful, A assigns label L2 for FEC PE1, installs forwarding state swapping labels L2 and L1 and advertises a binding for label L2 and FEC PE1 to its peer B, who will do similar processing. If the check is not successful, A would not advertise the FEC any further. In this fashion, the LSP setup proceeds in an orderly way from egress to ingress. Each LSR consults the IGP for two decisions: (a) whether to advertise a mapping for an FEC and (b) whether to use a label for forwarding.

Independent control. With independent control, each LSR assigns a label for FEC PE1 and advertises this binding independently of the peers. Each LSR uses the locally assigned label as its incoming label in the forwarding table. The outgoing label in the forwarding table is filled in when the LSR receives a label for PE1 from a peer lying directly on the IGP shortest path for prefix PE1. The LSRs use the IGP for just one

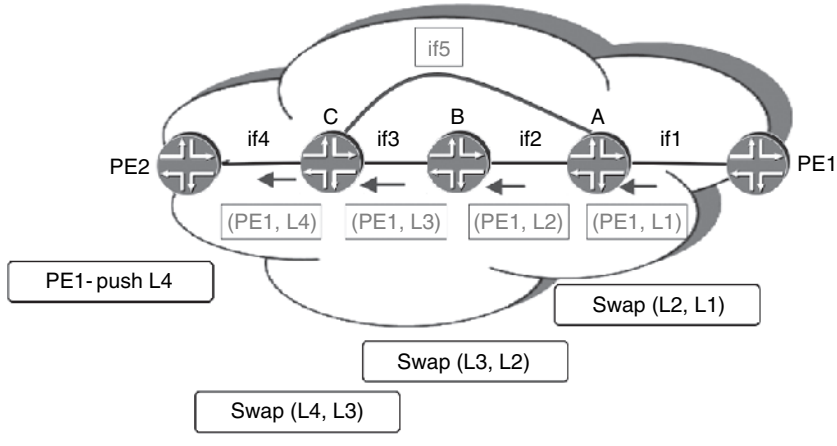

Figure 1.5 Different behavior for the ordered and independent label distribution modes

decision: whether to use a label for forwarding or not. The success of the LSP establishment depends on all LSR advertising labels for the same set of FECs. If LSR A were configured not to advertise a label for FEC PE1, the LSP to PE1 would never be established.

At this point, it is probably already clear that the default behavior regarding the choice of FECs that are advertised, which we discussed earlier in this section, is not an arbitrary one. With ordered control, the router who is the egress of the LSP decides which FECs to initiate LSPs for. Thus, a reasonable default behavior for an implementation performing ordered control is to advertise a mapping for the loopback address of the egress. With independent control, all routers in the network must advertise the same set of FECs. Thus, the reasonable thing for an implementation performing independent control is to advertise a mapping for all prefixes in the routing table. Another point to note is that when changing the default behavior via configuration, with ordered control the change is applied to one router only (the egress), while with independent control the change must be uniformly applied throughout the network. The requirement for a uniformly applied change is due to the independent operation of the routers in the network: unless they agree on the same set of FECs to advertise, LSPs will not establish end-to-end throughout the network, causing traffic blackholing. This situation is made worse by the fact that the protocol has no built-in mechanisms for detecting such misconfigurations.

The different behavior with regards to the propagation of labels has important implications regarding the setup of LSPs. With ordered control, the bindings must propagate from the egress to the ingress before the LSP is established and traffic can be forwarded on to it. If an application (such as a Layer 3 VPN) relies on the existence of the LSP, then it cannot forward traffic. This behavior is not limited to the initial setup of LSPs. The same dynamics apply when routing changes. With ordered control labels must propagate to the routers in the new IGP path, while with independent control the labels are already available on these routers. This, however, is not as bad as it looks: when routing changes, the IGP messages themselves must propagate and new routes computed, so the propagation of LDP labels is no worse than the propagation of IGP messages.

A more interesting scenario is a failure case where LDP cannot follow the IGP. Let us go back to the example in Figure 1.5. Assume that the interface if5 does not yet exist in the network. The LSP for FEC PE1 (the loopback of router PE1) establishes along the routers PE2–C–B–A–PE1. At this point, the operator decides to add the interface if5 and includes it in the IGP, but forgets to enable LDP on it. As a result, the IGP best path from router C for FEC PE1 is C–A–PE1.

With ordered control, LSR C notices that the label advertisement that it received for FEC PE1 from LSR B does not match the IGP best path,

withdraws its advertisement for FEC PE1 and removes its forwarding state. When LSR PE2 receives the withdrawal, it removes the forwarding state for FEC PE1. PE2 knows that the LSP is not operational and will not attempt to forward labeled traffic on it. With independent control, LSR C notices that the routing changed and that the outgoing label it installed in the forwarding table for FEC PE1 is no longer valid and removes the forwarding state for FEC PE1. PE2 does not change its forwarding state, since from its point of view the best path to PE1 is still through C. The net effect is that the LSP for PE1 is broken at point C, but PE2 is unaware of the failure. It will continue to send labeled traffic on this LSP and the traffic will be dropped at C. This type of silent failure is very problematic in a VPN environment, as we will see in later chapters. A solution to this issue is the scheme described in [LDP-IGP-SYNC], in which the IGP metric for a link is given a high value if LDP is not fully operational over the link. As described earlier, this scheme is also a solution to race conditions between LDP and the IGP.

Implementations supporting each of the two modes of operation can be and are deployed together in the same network [LDP-OP]. The key to interoperability is the fact that LSRs do not assume anything regarding the behavior of their peers, except consistent installation of the forwarding state following the IGP path.

Now that we have discussed the way LDP labels are distributed, let us look at an example of an LDP LSP. Figure 1.6 shows an LDP LSP whose egress point is router D. LDP forms a multipoint-to-point tree rooted at D, with each of the other routers as ingress points to the tree. In the same way, LDP also forms multipoint-to-point trees rooted at each of the other routers in the network, but these are not shown in the diagram for clarity. The numbers inside the boxes show the IGP metric on each link. The

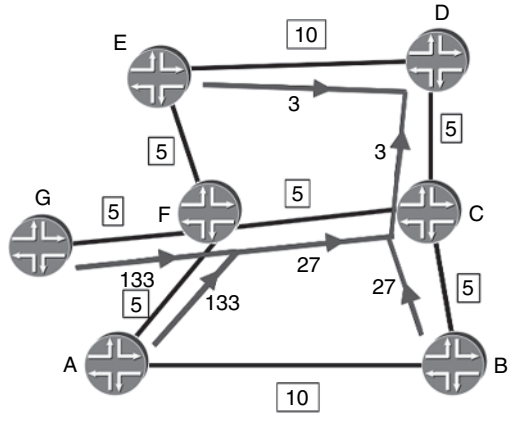

Figure 1.6 Inverted tree formed by LDP rooted at D

arrows show the direction of the data flow, and the number next to each arrow shows the LDP label used on that link for the LSP to D. It can be seen that the LSP path follows the best route as determined by the IGP. On any particular link, the label used to reach a particular destination router is the same, regardless of the origin of the packet. Thus, for example, on link F–C all packets whose destination is D have a label value of 27, regardless of whether they originated at G or A or F. Also, if per-platform label space is used, router C (for example) announces the same label value in order to reach D to all its neighbors, so all traffic passing via C to reach D has the same label value on all links into C. Hence traffic from B to D also has a label value of 27 on the B–C link. Note that in the example, D announces a label value of 3 to its neighbors. This label value of 3 is a special one called the 'Implicit NULL label' [RFC 3032]. This triggers PHP on C and E. Because of the special meaning associated with a label value of 3, an MPLS data packet could never have a header with a label value of 3. As already stated, the diagram only shows the tree rooted at D. In reality, there would be multiple overlapping trees, each rooted at a different router in the network. As a result, on any particular link various labels may be in use if multiple routers are reachable over that link.

As with the IGPs, typically LDP implementations install multiple forwarding table entries in Equal Cost Multi-Path (ECMP) situations. For example, in Figure 1.6, if the metric between E and D were 5 rather than 10, there would be two equal cost paths from F to D, F–E–D and F–C–D. Hence F installs two forwarding entries for D, one corresponding to each path. Traffic arriving at F for D is load-balanced over the two paths.

LDP key properties

Here is a summary of the key properties of LDP:

- Automatic discovery of peers. LDP uses discovery messages to find peer LSRs. This yields two important benefits:
  - *Ease of configuration*. The operator does not need to configure each peer individually. Adding a new LSR in the network requires configuration of the new LSR, but not of any of the other LSRs in the network (in contrast to RSVP). The automatic discovery built into the LDP protocol is one of the most compelling reasons for picking LDP as the label distribution protocol in networks where traffic engineering is not required.
  - *Session maintenance*. The amount of session state an LSR must maintain is proportional to the number of neighbors. In the absence of targeted peers, this number is constant, regardless of the size of the network.

- *Reliable transport*. LDP uses TCP as the transport protocol for all except the discovery messages. Once advertised, information does not need to be refreshed. Keepalive messages are sent periodically for session maintenance, but their number is proportional to the number of sessions, not to the amount of information that was exchanged over the session.
- *Extensible design*. LDP uses TLVs for passing information around. This has proven itself over and over as the protocol was extended over the years.
- *Reliance on the IGP*.[2] LDP relies on the IGP for the routing-related decisions. LDP-established LSPs follow the IGP shortest path and are influenced by changes in routing. During periods of network convergence, LDP LSPs are affected, and traffic may be looped or blackholed.
- *Liberal label retention and downstream unsolicited label distribution*. The labels are advertised to all peers and kept by the peers even if they are not actively used for forwarding. Thus LDP reacts quickly to changes in the IGP routing.

1.3.2.2 RSVP

Another scheme for distributing labels for transport LSPs is based on the Resource Reservation Protocol (RSVP). RSVP was invented before MPLS came into being, and was originally devised as a scheme to create bandwidth reservations for individual traffic flows in networks (e.g. a video telephony session between a particular pair of hosts) as part of the so-called 'int-serv' model. RSVP includes mechanisms for reserving bandwidth along each hop of a network for an end-to-end session. However, the original int-serv application of RSVP has fallen out of favor because of concerns about its scalability: the number of end-to-end host sessions passing across a service provider network would be extremely large, and it would not be desirable for the routers within the network to have to create, maintain and tear down state as sessions come and go.

In the context of MPLS, however, RSVP has been extended to allow it to be used for the creation and maintenance of LSPs and to create associated bandwidth reservations [RFC3209]. When used in this context, the number of RSVP sessions in the network is much smaller than in the case of the int-serv model because of the way in which traffic is aggregated into an LSP. A single LSP requires only one RSVP session, yet can carry all the traffic between a particular ingress and egress router pair, containing many end-to-end flows.

[2] Recall that the discussion in this section is for FECs that are IP addresses.

An RSVP-signaled LSP has the property that its path does not necessarily follow the path that would be dictated by the IGP. RSVP, in its extended form, has explicit routing properties in that the ingress router can specify the entire end-to-end path that the LSP must follow, or can specify that the LSP must pass through particular transit nodes. Here are a few consequences of the explicit routing properties of RSVP:

1. The path does not necessarily follow the IGP. The path can be computed to comply with different constraints that may not be taken into account when the IGP paths are computed. As such, RSVP-signaled LSPs are a key component of MPLS-based traffic engineering, enabling the network administration to control the path taken by traffic between a particular pair of endpoints by placing the LSP accordingly.
2. The path may be computed online by the router or offline using a path computation tool. In the case of online computation, typically only the ingress router needs to be aware of any constraints to be applied to the LSP. Moreover, use of the explicit routes eliminates the need for all the routers along the path to have a consistent routing information database and a consistent route calculation algorithm.
3. The path is not restricted to a single IGP instance. As long as a path is specified in some way, RSVP is not restricted to a single IGP instance (so, for example, is not confined to one AS). In contrast, LDP is dependent on the IGP, so although LDP LSPs can cross from one IGP area or level to another, they cannot cross from one AS to another, since different ASs run separate IGPs.[3]
4. An LSP can be signaled in such a way that its path can only be changed by the head end. This is in contrast to LDP, where each LSR updates its forwarding state independently of all other LSRs as it tracks the IGP state. This property is very important in the context of traffic protection schemes such as fast reroute, discussed in detail in the Protection and Restoration chapter of this book (Chapter 3). Fast-reroute schemes involve each router along the path of an LSP computing a local repair path that bypasses a failure in the downstream link or downstream neighbor node. Traffic sent on the LSP is guaranteed to reach the router where the local repair path has been set up, since the routers do not change their forwarding state after a failure (this again is in contrast to the looping that may happen with LDP following a failure).

The creation of an RSVP-signaled LSP is initiated by the ingress LER. The ingress LER sends an RSVP Path message. The destination address of the Path message is the egress LER. However, the Path message has

[3] A workaround is to leak the addresses corresponding to the LDP FECs between the IGPs in the two ASs, but this is cumbersome and is only used in situations where the ASs involved belong to the same owner.

the Router Alert option set so that transit routers can inspect the contents of the message and make any necessary modifications.

Here are some of the objects contained in a Path message:

1. *Label Request Object*. Requests an MPLS label for the path. As a consequence, the egress and transit routers allocate a label for their section of the LSP.
2. *Explicit Route Object (ERO)*. The ERO contains the addresses of nodes through which the LSP must pass. If required, the ERO can contain the entire path that the LSP must follow from the ingress to the egress.
3. *Record Route Object (RRO)*. RRO requests that the path followed by the Path message (and hence by the LSP itself once it is created) be recorded. Each router through which the Path message passes adds its address to the list within the RRO. A router can detect routing loops if it sees its own address in the RRO.
4. *Sender TSpec*. TSpec enables the ingress router to request a bandwidth reservation for the LSP in question.

In response to the Path message, the egress router sends an Resv message. Note that the egress router addresses the Resv message to the adjacent router upstream, rather than addressing it directly to the source. This triggers the upstream router to send a Resv message to its upstream neighbor and so on. As far as each router in the path is concerned, the upstream neighbor is the router from which it received the Path message. This scheme ensures that the Resv message follows the exact reverse path of the Path message. Figure 1.7 illustrates the Path and Resv message exchange along the path of an LSP.

Here are some of the objects contained in an Resv message:

1. *Label Object*. Contains the label to be used for that section of the LSP. For example, in Figure 1.7 when the Resv message is sent from the egress router Z to the upstream neighbor Y, it contains the label value that Y must use when forwarding traffic on the LSP to Z. In turn, when Y sends the Resv message to X, it overwrites the Label Object with the label value that X must use when forwarding traffic on the LSP to Y. In this way, for the LSP in question, Y knows the label with which traffic arrives at Y and the label and outgoing interface that it must use to forward traffic to Z. It can therefore install a corresponding label swap entry in its forwarding table.
2. *Record Route Object*. Records the path taken by the Resv message, in a similar way to the RRO carried by the Path message. Again, a router can detect routing loops if it sees its own address in the Record Route Object.

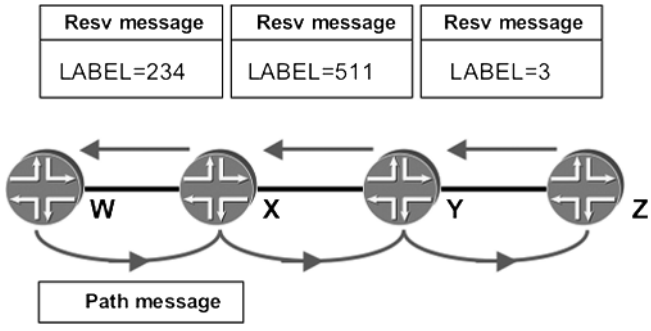

Figure 1.7 Illustration of the RSVP Path and Resv message exchange

As can be seen, RSVP Path and Resv messages need to travel hop-by-hop because they need to establish the state at each node they cross, e.g. bandwidth reservations and label setup.

As a consequence of the scheme described above, an RSVP-signaled LSP only requires configuration at the ingress router. In typical implementations, properties of the LSP and the underlying RSVP session, such as the ERO and RRO and requested bandwidth, can be viewed on any router along the path of the LSP since that information is known to all routers along the path.

RSVP requires a periodic exchange of messages once an LSP is established in order to maintain ('refresh') its state. This can be achieved by periodically sending Path and Resv messages for each active LSP. If a router does not receive a certain number of consecutive Path or Resv messages for a particular LSP, it regards the LSP as no longer required and removes all states (such as forwarding entries and bandwidth reservations) pertaining to that LSP. The processing overhead of such a scheme can become a scaling concern for a router maintaining a very large number of LSPs. In order to address this, the 'Refresh Reduction Extensions' to RSVP were devised to reduce this overhead. These include a Summary Refresh Extension that allows multiple RSVP sessions (and hence multiple LSPs) to have their state refreshed by a single message sent between RSVP neighbors for refresh interval [RFC2961].

RSVP has an optional node failure detection mechanism, in which hello messages are sent periodically between RSVP neighbors. Without this mechanism, a node might only become aware of the failure of a neighbor through the timeout of RSVP sessions, which can take a relatively long time.

Note that there is no concept of ECMP in RSVP as there is in LDP. A particular RSVP-signaled LSP follows a single path from ingress to egress. If, in performing the path computation, the ingress router finds that there are multiple potential paths for an LSP that have equal merit, it chooses one of those paths for the LSP and signals for its creation via

RSVP. Hence, once traffic has entered an RSVP-signaled LSP, there is no splitting and merging of traffic as sometimes occurs in the LDP case. On the other hand, if the ingress router has multiple RSVP-signaled LSPs to a particular egress router, it can load-balance the traffic across those LSPs. Some implementations allow the load-balancing to be weighted according to the bandwidth reservation of the LSPs.

In some cases, a network may only have a handful of RSVP-signaled LSPs, as a tactical way of controlling traffic flows around particular hotstops in the network. In those situations, RSVP-signaled LSPs would be created between certain pairs of endpoints to achieve this aim. In other networks, the reason for deploying RSVP-signaled LSPs might be in order to make use of fast reroute, in which case the administrator may choose to fully mesh the PEs in the network with RSVP-signaled LSPs.

By way of summary, here are the key properties of RSVP:

- Explicit routing. The ingress LER has control over the path taken by the LSP, either by specifying the entire path or by specifying particular nodes that the LSP must pass through. As a consequence, RSVP lends itself to traffic engineering and traffic protection schemes that operate independently of, and faster than, the IGP.
- Periodic message exchange is required to renew the state of an LSP, although the RSVP Refresh Reductions reduce this overhead.
- The amount of session state on a node is proportional to the number of LSPs traversing the node. This tends to grow as the network grows (assuming a high degree of meshing of RSVP-signaled LSPs).

1.3.2.3 RSVP and LDP comparison

A frequently asked question is whether LDP or RSVP is the better protocol to use in a deployment. Let us compare the two protocols with regard to the factors that affect the choice of which to use:

1. Ease of configuration:
   (a) *Initial configuration*. LDP has the advantage that it is easy to configure, only requiring one line of configuration in some implementations, to allow the protocol to run on a particular interface. RSVP, on the other hand, requires explicit configuration of the LSPs on the ingress router. Each router must know all other routers to which it must establish LSPs.
   (b) *Incremental configuration when new edge devices are added*. For LDP, only the new device must be configured. For RSVP, adding a new router to the edge means configuring LSPs to it from all the existing routers, potentially requiring configuration changes on all other edge routers in the network.

There are currently moves to reduce the configuration effort when using RSVP. One scheme is an automatic meshing capability, where each edge router in the network automatically creates an RSVP-signaled LSP to the other edge routers in the network. Another is an autobandwidth capability, where the bandwidth reservation for an LSP changes in accordance with the volume of traffic using that LSP. Used in combination, the configuration effort would not be very different to that associated with LDP. Such schemes may not help in all cases, however, e.g. when each LSP has particular constraints associated with it or requires a fixed bandwidth reservation rather than one that dynamically varies.

2. Scalability:

   (a) *Control plane sessions*. For LDP, each router must maintain a number of sessions equal to the number of LDP neighbors. For RSVP, the number of sessions is equal to the total number of LSPs that the router is involved with (whether in the role of ingress, transit or egress router). For a fully meshed topology, the total number of LSPs in the network is of order *N*-squared in the RSVP case, where *N* is the number of edge routers, because each edge router has an LSP to each of the other edge routers but is proportional to *N* in the LDP case, because each edge router is the egress for an LDP multipoint-to-point tree having every other edge router as an ingress point.
   (b) *State maintenance*. LDP sends periodic keepalive and hello messages, but only for a limited and constant number of neighbors/sessions. RSVP must refresh all sessions for the LSPs traversing a router, a number over which it has no control. RSVP refresh reduction reduces the number of RSVP messages that have to be created and sent in order to refresh the sessions; however, the router still needs to track the state of each session.
   (c) *Forwarding state*. LDP maintains the forwarding state for all FECs in the network. By nature of the protocol each FEC is reachable from every point in the network. The ability of LDP to support ECMP means that often more than one path is maintained. RSVP, on the other hand, only keeps the state for the LSPs traversing it, and potentially their protection paths.

For practical purposes, the above considerations may not be of practical importance unless one has a very large number of routers that need to be fully meshed with RSVP-signaled LSPs, resulting in an unsustainably large number of LSPs to be maintained by routers in the core of the network. In those cases, either the LDP over RSVP or the LSP hierarchy schemes described later in this section can be used.

3. *Features supported*. Currently, only RSVP supports traffic engineering and fast reroute.

From the above analysis it should come as no surprise that if the traffic engineering or fast-reroute properties offered by RSVP are not required, LDP is almost always chosen. Let us take a closer look at the choice of protocol in the context of the application for which the MPLS connectivity is required:

1. *L3VPN*. These services often do not have stringent SLAs in terms of outage time in the event of a link failure and although they may offer several Diff-Serv traffic classes, none of the traffic classes have associated bandwidth reservations through the core. The main considerations in this case are ease of management and provisioning. Therefore, to date, LDP has received wider deployment than RSVP in such networks.
2. *Migration of Layer 2 services to MPLS networks*. Emulation of services such as ATM and Frame Relay over MPLS networks often requires tangible bandwidth guarantees. For example, if a service provider offers a particular access rate at a particular class of service between two access points in the network, it is necessary to ensure that the bandwidth between those points is reserved and uncontended. In addition to the bandwidth guarantees, Layer 2 services require fast restoration following a link failure. Due to its fast reroute and traffic engineering capabilities (and in particular DiffServ Aware Traffic Engineering), RSVP is better suited than LDP in such deployments.
3. *Services requiring fast restoration, such as voice services*. In some cases, there may be no TE requirement, because link utilization is low and bandwidth plentiful. However, fast-reroute capabilities may still be required, due to the nature of the service (e.g. voice). RSVP is the only protocol that supports fast restoration today. To cater for service providers (SPs) that require faster restoration times but do not require traffic engineering, there are moves to improving the convergence time of traffic traveling down LDP-signaled LSPs. In some cases, it is advantageous to use a combination of RSVP and LDP-signaled LSPs.

In many deployments, each Point-of-Presence (PoP) consists of several access routers and one or two core facing routers. The SP may wish to use RSVP for its traffic engineering properties in the core, but has no need for traffic engineering within the PoP. Similarly, there may be a need for fast reroute in the core but not within the PoP infrastructure, on the premise that intra-PoP link failure is relatively rare.

In these cases, the SP can use LDP within the PoPs and RSVP-signaled LSPs in the core. This is illustrated in Figure 1.8.

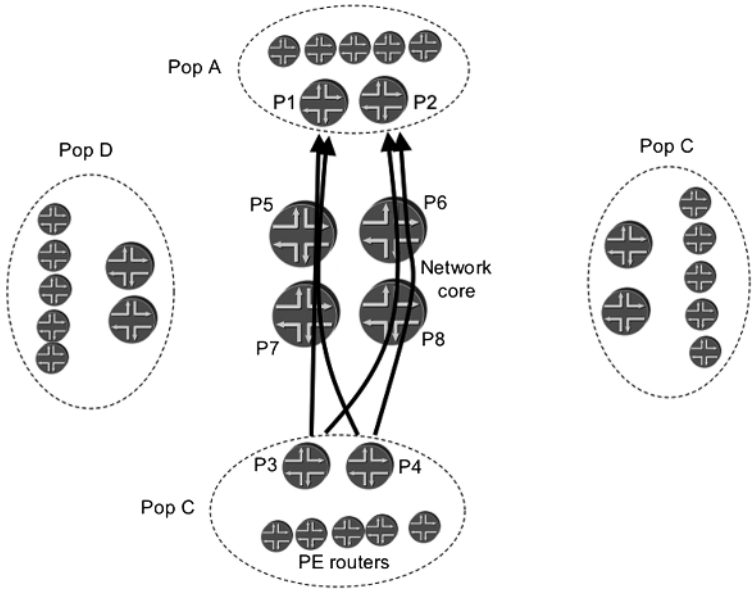

Figure 1.8 Creating an RSVP mesh between core-facing P routers in each PoP

In Figure 1.8, each PoP has five LERs and two core-facing LSRs. Each core-facing LSR has an RSVP-signaled LSP to each of the core-facing LSR in the other PoPs. In the figure, we show only the RSVP-signaled LSPs from PoP C to PoP A for clarity. Targeted LDP sessions are created between the ingress and egress routers of each RSVP-signaled LSP. A targeted LDP session allows LDP labels to be exchanged between routers even if they are not directly connected to each other so that LDP labels are exchanged without involving the transit routers of the RSVP-signaled LSPs. For example, there would be a targeted LDP session between P3 and P1, and the routers in the core of the network (P5, P6, P7 and P8) would not be involved in this session. Let us look at the impact that the LDP over RSVP scheme has on the total number of RSVP-signaled LSPs in the network. If the number of core-facing routers in the network is X and the number of edge routers in the network is Y, then the number of RSVP-signaled LSPs is reduced from $Y\ (Y-1)$ to $X\ (X-1)$. This could be a large reduction if the ratio Y to X is large. For example, consider a network that has 30 PoPs, each containing two core-facing routers and five edge routers. In the case where the edge routers are fully meshed with RSVP-signaled LSPs, there would be 22 350 (i.e.150×149) RSVP-signaled LSPs in the network. In the case where only the two core-facing routers in each PoP are fully meshed, there would be a total of 3480

(i.e. 60×58) RSVP-signaled LSPs in the network.[4] This is almost an order of magnitude smaller than the full mesh case. The smaller number of LSPs means a lighter load on the protocols and the routers. This, in itself, is only of practical consequence if the load in the fully meshed edge router case is unsustainably high. More importantly, fewer LSPs means easier provisioning and management from the operator's point of view.

The LDP over RSVP process is illustrated in more detail in Figure 1.9, which shows a cross-section through the edge and core of a network. Routers A, B and C are within the same PoP. Routers F, G and H are within another PoP. D and E are core routers. LDP is used within the PoPs. In the network, the core-facing routers in the PoPs are fully meshed with RSVP-signaled LSPs. Hence there is a pair of RSVP-signaled LSPs between C and F (one in each direction). Also, there are targeted LDP sessions between the core-facing routers in each PoP, i.e. between C and F in the diagram. The targeted LDP session allows C and F to directly exchange labels for the FECs associated with the edge routers in their respective PoPs even though C and F are not directly connected. For example, C learns the label from F to use when forwarding traffic to H. Routers D and E are not involved in the LDP signaling process and do not store LDP labels.

Let us consider the transport of packets arriving into the network at router A and leaving the network at router H. The forwarding plane operation is as follows: ingress router A pushes a label which is learnt via LDP. In the example, the label value is L1, and is the label associated with H, the egress point of the packet. Router B swaps the label for one having the value L2. Router C is the ingress router for the RSVP-signaled LSP across the core. C swaps the existing label L2 for a label value L3 that it learnt via the targeted LDP session with F. Also, it pushes on to

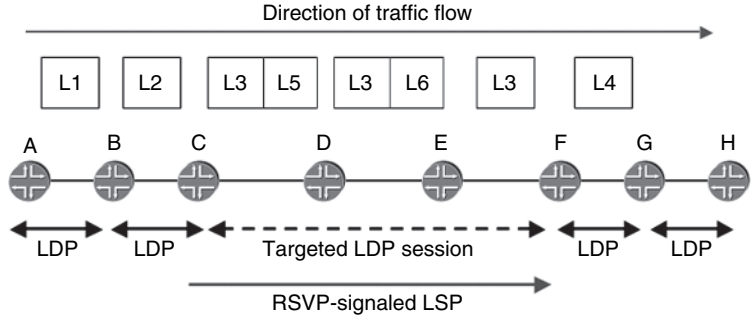

Figure 1.9 LDP over RSVP forwarding

[4] This calculation assumes that the core-facing router in each PoP does not need an RSVP-signaled LSP to the other core-facing router in the same PoP.

the packet a label of value L5 learnt via RSVP. Hence, at this point, the label stack consists of an outer label of value L5 and an inner label of value L3. The core routers D and E are only aware of the RSVP-signaled LSP and hence only carry out operations on the outer label. D swaps the outermost label of value L5 for a label having value L6. Note that the underlying label having value L3 is left untouched. If PHP is in use, router E pops the label learnt via RSVP, thus exposing the label, L3, learnt via LDP. Router F swaps the LDP label for one having value L4. If PHP is in use, router G pops the label, exposing the header of the underlying packet. This could be an IP header or could be another MPLS header, e.g. a VPN label.

In cases where the properties brought by RSVP are required from edge to edge, the above LDP over RSVP scheme is not suitable. However, in the case of very large networks, it may not be feasible either to fully mesh all the edge routers with RSVP-signaled LSPs because of the resulting amount of the RSVP state in the core of the network. The concept of LSP hierarchy [RFC 4206] was introduced to solve this problem. In this scheme, a layer of routers is fully meshed with RSVP-signaled LSPs. The layer is chosen such that the number of routers involved in the mesh is less than the number of edge routers. For example, as with the LDP over RSVP scheme discussed earlier, the routers chosen might be the core-facing routers within each PoP. The edge routers are also fully meshed with RSVP-signaled LSPs which are nested within the LSPs between the core-facing routers.[5] The LSPs in the core of the network are called forwarding adjacency (FA) LSPs. Referring again to Figure 1.8, in the context of LSP hierarchy, the LSPs between P1, P2 and P3 and P4 are the FA LSPs. Each LER would have an RSVP-signaled LSP to each other LER in the network, which would be tunneled in one of the FA-LSPs in order to cross the core. In this way, routers in the heart of the network (P5, P6, P7 and P8 in the figure) only have to deal with the session state corresponding to the core LSPs and are unaware of the fact that LSPs from LER to LER are nested within them.

The LSP hierarchy concept is illustrated in more detail in Figure 1.10. The diagram shows six LERs, three in each of two PoPs. P1 is a core-facing router in one PoP and P3 is a core-facing router in the other PoP. The diagram shows an RSVP-signaled LSP between P1 and P3. Using LSP hierarchy, edge-to-edge LSPs between the LERs in the two PoPs can be nested within the core LSP between P1 and P3. For example, there is an LSP between PE1 and PE4, another between PE2 and PE5 and so on. However, P2 in the core of the network is unaware of the existence of these LSPs and is only involved in the maintenance of the core LSP. This is because the RSVP messages associated with the edge-to-edge LSPs pass

[5] Note that, as a consequence, the use of the LSP hierarchy does not solve the issue of the overhead of configuring a full mesh of RSVP-signaled LSPs.

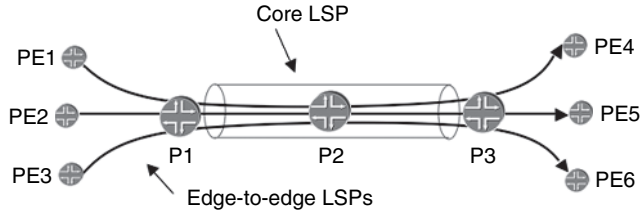

Figure 1.10 LSP hierarchy

directly between P1 and P3 without being processed by the control plane of P2. Note that in the data plane, the label operations are analogous to those in the LDP over RSVP case that we showed in Figure 1.9. The ingress router of the FA-LSP pushes a label corresponding to the FA-LSP onto the existing label stack. This label is swapped at each hop of the FA-LSP, leaving the labels underneath untouched and is then typically popped at the penultimate router of the FA-LSP.

1.3.2.4 BGP label distribution

The third type of label distribution also relies on a preexisting protocol, BGP. BGP has support for multiple address families, which make it straightforward to define and carry new types of reachability information and associated attributes. Thus, by adding a new address family to BGP, it is possible to advertise not just a prefix but also one or more labels associated with the prefix. In the Hierarchical and Inter-AS VPNs chapter of this book (Chapter 9), we will see that this capability is essential in the context of inter-AS MPLS/VPNs. The chapter describes several solutions in which BGP is used to:

(a) distribute the 'inner' labels (VPN labels) required by the egress LER to identify the service and service instance that the packet belongs to and/or

(b) distribute the outer label required to transport a packet to the appropriate egress LER.

The reasons for picking BGP as the protocol for the solution are discussed in detail in the Hierarchical and Inter-As VPNs chapter (Chapter 9). At this point, let us see some of added benefits of using BGP for label distribution:

- The ability to establish LSPs that cross AS boundaries. An example of where this is required is an MPLS-based VPN service having attachment points within multiple providers. In this case, it is necessary to distribute labels pertaining to LER reachability, so that the transport label required to reach a LER in another AS is known. BGP is a

protocol that is used today to convey reachability information across AS boundaries; therefore it can easily convey label information across AS boundaries.

- Reduction in the number of different protocols running in the network. Rather than deploying an entirely new protocol, reuse one of the existing protocols to provide one more function.
- Reuse of existing protocol capabilities. BGP supports a rich set of attributes that allow it to filter routing information, control the selection of exit points, prevent loops, etc. All these capabilities are readily available when label information is distributed along with a prefix.

BGP label distribution is also used in the context of the 6PE scheme to enable transport of IPv6 over an IPv4 MPLS core. This is discussed in the next section.

1.3.3 Transport of IPv6 over an IPv4 MPLS core

Increasingly, service providers are seeing the need to carry IPv6 traffic as well as IPv4 traffic across their networks. As for IPv4 traffic, the IPv6 traffic can be divided into two main categories:

(i) *Public IPv6 traffic (or 'IPv6 Internet' traffic)*. In this case, the requirement for the service provider is to transport IPv6 packets between IPv6 users across the public Internet infrastructure. In some cases, packets might be transported between users attached to the same service provider's network, but more typically the task of the service provider is to transport IPv6 packets between a service provider customer and an IPv6-enabled peering exchange, for hand-off to another service provider.

(ii) *Private IPv6 traffic*. In this case, the requirement is to provide a VPN service, to enable IPv6 traffic to be transported between a customer's sites while maintaining separation and privacy from other customers.

In this section, we will examine case (i), the public IPv6 case, in more detail. Case (ii), private IPv6 traffic, will be discussed in the Advanced Topics in Layer 3 BGP/MPLS VPNs chapter.

The service provider has the following choices in terms of how to carry the IPv6 traffic across the network core:

1. Turn on IPv6 forwarding and an IPv6-enabled IGP on all the routers in the network and send the packets in native IPv6 form.
2. Create a mesh of tunnels (such as GRE tunnels) between the PE routers in the network. Thus, the IPv6 packets can be encapsulated in IPv4, avoiding the need to turn on IPv6 in the core of the network.

3. Use MPLS LSPs between the PE routers in the network to carry the IPv6 packets.

Option 1 is of interest to service providers that already carry IPv4 internet traffic across the core in native IPv4 form, because the IPv6 traffic is carried in an analogous way. However, in some cases, a service provider may have core routers that are not capable of running IPv6 or there may be a reluctance to turn on IPv6 on the core routers.

In contrast, Option 2 avoids the need to turn on IPv6 on the core routers, because the IPv6 packets are encapsulated inside IPv4 packets. The issue with this scheme, however, is that typically it involves manual configuration of the tunnels and so has a high operational overhead.

Option 3 is attractive to service providers who already use MPLS LSPs to carry their IPv4 internet traffic, as it allows the same LSPs to be used to carry the IPv6 internet traffic too. The configurational overhead is much less than for Option 2.

Let us examine Option 3 in more detail. A scheme called '6PE' [RFC 4798] has been devised to cater for this scenario. The premise behind the scheme is that the core routers in the network do not support IPv6, so only the LERs need to support IPv6 forwarding and an IPv6 protocol. The LSPs used to transport the packets are signaled using IPv4 and can be the same LSPs that are used to transport IPv4 traffic and other traffic such as Layer 2 traffic.[6] Figure 1.11 illustrates the infrastructure required for the 6PE scheme.

The service provider's LERs routers each have an eBGP session running over IPv6 to the attached CE routers. Similarly, the service provider's

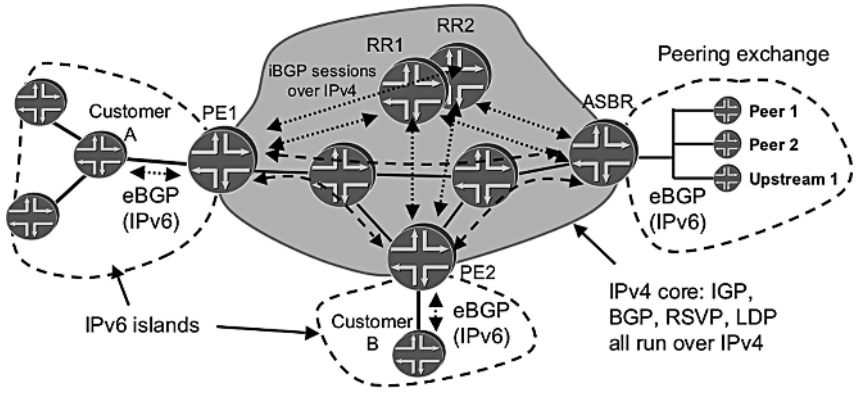

Figure 1.11 Carrying IPv6 traffic across the core using the 6PE scheme. Dotted lines denote BGP sessions. Dashed lines denote MPLS LSPs

[6] Although the IETF has defined schemes for signaling LSPs using IPv6, these are not supported by most implementations today.

peering router has eBGP sessions running over IPv6 with peers and upstream providers. Within the core of the network, shaded in grey, the addressing and the IGP are IPv4 based. The LERs in the network are meshed with LSPs that are signaled using IPv4. The iBGP sessions for exchanging routes between the LERs also run over IPv4.

In order to discuss the label operations associated with the 6PE scheme, let us re-examine the LSP in Figure 1.7. The LSP in the figure happens to have been signaled using RSVP, but the same holds if it had been signaled using LDP. Imagine if one simply encapsulated the IPv6 packet into the LSP shown in the figure. Between X and Y, the packet would have a label value of 511. At Y, the label would be popped, since PHP is in operation. Although the use of PHP is not mandatory, in practice it is used in the majority of MPLS deployments. This would give rise to the issue that a bare IPv6 packet would be exposed on router Y. However, the premise behind the 6PE scheme is that the P routers do not support IPv6. If this is the case for router Y, then Y would not know how to set the appropriate protocol type in the Layer 2 header before forwarding the packet on the link to Z. For example, if an Ethernet link is being used between Y and Z, router Y would need to set the Ethertype on the Ethernet frame to the value assigned for IPv6 payloads. In order to overcome this problem, the 6PE solution makes use of an additional label to ensure that the IPv6 packet is not exposed to the penultimate router. This is illustrated in Figure 1.12.

There is an MPLS LSP from PE1 to PE2, signaled by LDP or RSVP. On PE1, an MPLS header having label value Y is pushed onto the IPv6 packet. On top of that, another MPLS header having label value X is pushed onto the packet. The label value X is the one signaled by LDP or RSVP. At P1, the outer label value is swapped for a label having value W. At P2, the outer label is popped, exposing the inner label having value A. The packet is forwarded with this label to PE2. But how does PE1 know what label value is required for the inner label? The answer is to use Multi-Protocol (MP) BGP. In this way, when PE2 advertises its IPv6

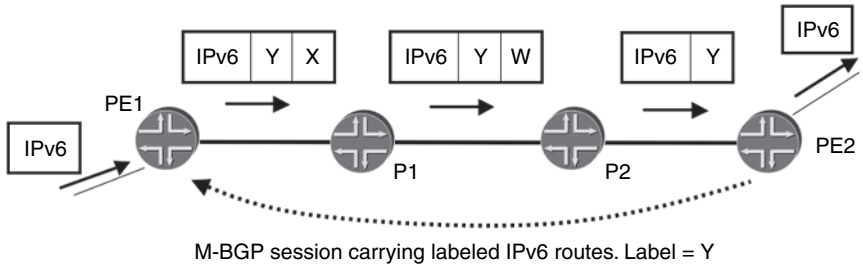

Figure 1.12 Using an extra label in the 6PE scheme

prefixes in BGP, it also advertises the label value associated with them. The Address Family Indicator (AFI) used for this advertisement has a value of 2, signifying IPv6. The Subsequent Address Family Indicator (SAFI) used has a value of 4, signifying a labeled route. The same BGP session can also be used to advertise IPv4 prefixes, without a label. The same LSP can be used to carry IPv4 traffic as is used to carry IPv6 traffic between PE1 and PE2. At each hop, the label stack for an IPv6 packet would have one extra label compared to the label stack for an IPv4 packet.

Note that 6PE is not a VPN scheme. If the requirement is to provide a VPN service capable of transporting a customer's IPv6 packets, the scheme discussed in the Advanced Topics in Layer 3 BGP/MPLS Virtual Private Networks chapter should be used.

1.4 CONCLUSION

We have started this chapter by looking at the original goals of the MPLS Working Group back in 1997. As is often the case for successful technologies, MPLS has become a key component in the development of new applications that were not envisioned at the time MPLS started out. The following chapters take a closer look at many of the innovations made possible by MPLS.

1.5 REFERENCES

[Davie Rekhter] B. Davie and Y. Rekhter, *MPLS: Technology and Applications*, Morgan Kaufmann, 2000

[Doyle Kolon] J. Doyle and M. Kolon (eds), *Juniper Networks Routers: The Complete Reference*, McGraw-Hill, 2002

[LDP-IGP-SYNC] M. Jork, A. Atlas and L. Fang, *LDP IGP Synchronization*, draft-jork-ldp-igp-sync-03.txt (work in progress)

[LDP-OP] L. Andersson, I. Minei and B. Thomas, *Experience with the LDP protocol*, draft-minei-ldp-operational-experience-00.txt (work in progress)

[RFC 3032] E. Rosen, D. Tappan, G. Fedorkow, Y. Rekhter, D. Farinacci, T. Li and A. Conta, *MPLS Label Stack Encoding*, RFC 3032, January 2001

[RFC 4206] K. Kompella and Y. Rekhter, *LSP Hierarchy with Generalized MPLS TE*, RFC 4206, October 2005

[MPLS97] Original problem statement for the IETF MPLS Working Group, http://www.ietf.org/proceedings/97apr/97apr-final/xrtftr90.htm

[MPLS ALLI] T. Walsh and R. Cherukuri, *Two reference models for MPLS control plane interworking*, MPLS/FR Alliance Technical Committee document mpls2005.050.00, March 2005

[MPLS WG] IETF MPLS Working Group, http://ietf.org/html.charters/mpls-charter.html

[Osborne Simha] E. Osborne and A. Simha, *Traffic Engineering with MPLS*, Cisco Press, 2002

[RFC2961] L. Berger, D. Gan, G. Swallow, P. Pan, F. Tommasi and S. Molendini, *RSVP Refresh Overhead Reduction Extensions*, RFC2961, April 2001

[RFC3031] E. Rosen, A. Viswanathan and R. Callon, *Multiprotocol Label Switching Architecture*, RFC 3031, January 2001

[RFC5036] L. Andersson, I. Minei, B. Thomas (Eds), *LDP Specification*, RFC 5036, October 2007

[RFC3209] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan and G. Swallow, *RSVP-TE: Extensions to RSVP for LSP Tunnels*, RFC3209, December 2001

[RFC3945] E. Mannie, *Generalized multi-protocol label switching (GMPLS) architecture*, RFC 3945, October 2004

[RFC 4798] J. De Clercq, D. Ooms, S. Prevost, F. Le Faucheur, *Connecting IPv6 Islands over IPv4 MPLS Using IPv6 Provider Edge Routers (6PE)*, RFC 4798, February 2007

1.6 FURTHER READING

[LDP-MTU] B. Black and K. Kompella, *Maximum Transmission Unit Signaling Extensions for the Label Distribution Protocol*, RFC3988, January 2005

[RFC3478] M. Leelanivas, Y. Rekhter and R. Aggrawal, *Graceful Restart Mechanism for Label Distribution Protocol*, RFC3478, February 2003

1.7 STUDY QUESTIONS

1. List the fields in an MPLS header and describe their function.
2. Describe the two different schemes by which the Diff-Serv Per-Hop Behavior (PHB) can be inferred for an LSP.

3. Describe the differences between ordered and independent control modes for LDP LSP creation.
4. List some of the differences between LDP and RSVP.
5. A network has 100 LERs. How many LSPs are there in the network in total if it is required to fully mesh the LERs with RSVP-signaled LSPs?
6. A service provider wishes to carry IPv6 Internet traffic. The edge routers in the network support IPv6, but the core routers do not. List the methods by which the service provider can carry the IPv6 traffic across the network.
7. Describe the protocol machinery required for the 6PE scheme.