

GUIDE TO MPLS

By J Doyle

The key thing to remember about MPLS (Multiprotocol Layer Switching) is that it's a technique, not a service, so it can be used to deliver anything from IP VPNs to Metro Ethernet services, or even to provision optical services. So although carriers build MPLS backbones, the services that users buy may not be called MPLS. They could be called anything from IP VPN to Metro Ethernet, or whatever the carriers' marketing departments dream up next.

NOTE: In computer networking and telecommunications, Multi Protocol Label Switching (MPLS) is a data-carrying mechanism that belongs to the family of packet-switched networks. MPLS operates at an OSI Model layer that is generally considered to lie between traditional definitions of Layer 2 (Data Link Layer) and Layer 3 (Network Layer), and thus is often referred to as a "Layer 2.5" protocol. It was designed to provide a unified data-carrying service for both circuit-based clients and packet-switching clients which provide a datagram service model. It can be used to carry many different kinds of traffic, including IP packets, as well as native ATM, SONET, and Ethernet frames.

A number of different technologies were previously deployed with essentially identical goals, such as frame relay and ATM. MPLS technologies have evolved with the strengths and weaknesses of ATM in mind. Many network engineers agree that ATM should be replaced with a protocol that requires less overhead, while providing connection-oriented services for variable-length frames. MPLS is currently replacing some of these technologies in the marketplace. It is highly possible that MPLS will completely replace these technologies in the future. Thus aligning these technologies with current and future technology needs.

In particular, MPLS dispenses with the cell-switching and signaling-protocol baggage of ATM. MPLS recognizes that small ATM cells are not needed in the core of modern networks, since modern optical networks (as of 2008) are so fast (at 40 Gbit/s and beyond) that even full-length 1500 byte packets do not incur significant real-time queuing delays (the need to reduce such delays — e.g., to support voice traffic — was the motivation for the cell nature of ATM).

At the same time, MPLS attempts to preserve the traffic engineering and out-of-band control that made frame relay and ATM attractive for deploying large-scale networks.

While the traffic management benefits of migrating to MPLS are quite valuable (better reliability, increased performance), there is a significant loss of visibility and access into the MPLS cloud for IT departments.

The fundamental concept behind MPLS is that of labeling packets. In a traditional routed IP network, each router makes an independent forwarding decision for each packet based solely

GUIDE TO MPLS

By J Doyle

on the packet's network-layer header. Thus, every time a packet arrives at a router, the router has to "think through" where to send the packet next.

With MPLS, the first time the packet enters a network, it's assigned to a specific forwarding equivalence class (FEC), indicated by appending a short bit sequence (the label) to the packet. Each router in the network has a table indicating how to handle packets of a specific FEC type, so once the packet has entered the network, routers don't need to perform header analysis. Instead, subsequent routers use the label as an index into a table that provides them with a new FEC for that packet.

This gives the MPLS network the ability to handle packets with particular characteristics (such as coming from particular ports or carrying traffic of particular application types) in a consistent fashion. Packets carrying real-time traffic, such as voice or video, can easily be mapped to low-latency routes across the network — something that's challenging with conventional routing. The key architectural point with all this is that the labels provide a way to "attach" additional information to each packet — information above and beyond what the routers previously had.

Layer 2 or Layer 3

There's been a lot of confusion over the years about whether MPLS is a Layer 2 or Layer 3 service. But MPLS doesn't fit neatly into the OSI seven-layer hierarchy. In fact, one of the key benefits of MPLS is that it separates forwarding mechanisms from the underlying data-link service. MPLS can be used to create forwarding tables for ATM or frame relay switches (using the existing ATM or DLCI header) or for plain old IP routers by appending MPLS tags to IP packets.

The bottom line is that network operators can use MPLS to deliver a wide variety of services. The two most popular implementations of MPLS are layer 3 BGP/MPLS-VPNs (based on RFC 2547) and Layer 2 (or pseudowire) VPNs. RFC 2547 VPNs have been implemented by most of the major service providers, including AT&T, Verizon, BT and many others. The fundamental characteristics of a 2547 is that traffic is isolated into MPLS-VPNs as it enters the network.

Interior routers have no knowledge of IP information beyond the label-only base forwarding decisions on the MPLS label. BGP is used by edge routers to exchange knowledge of VPNs, thus enabling service providers to isolate traffic from multiple customers or even the Internet over a shared backbone.

There are several flavors of layer 2 MPLS services, but what they have in common is that a Layer 2 packet (or ATM cell or frame relay frame) is encased in an MPLS header and forwarded through the MPLS core. When it reaches the other side, the packet's labels are removed, and the packet that arrives at the ultimate destination exactly where it entered the

GUIDE TO MPLS

By J Doyle

MPLS network. Thus, Layer 2 MPLS services effectively extend services such as Ethernet or frame relay across an IP WAN.

What are the different types of MPLS?

The version of MPLS that's generally used to encapsulate connection-oriented frame relay and ATM services is called pseudo Wire Edge to Edge Emulation (PWE3). PWE3 defines point-to-point tunnels across the MPLS backbone, and thus works well for circuit-oriented networking protocols. PWE3 can also be used to support connectionless LAN protocols, but it's not the preferred solution.

For connectionless protocols (primarily Ethernet) there's a different specification, called virtual private LAN service (VPLS). VPLS addresses some of the specific challenges with extending Ethernet across the metropolitan area or WAN, most notably scalability and availability. Another emerging spec is the ITU's transport-MPLS (T-MPLS), which is designed to simplify deployment of Ethernet services.

It's worth noting that MPLS isn't the only game in town when it comes to Ethernet services, though. Several vendors —including Nortel, Extreme and Siemens — are promoting an alternative approach called Provider Backbone Transport, or PBT, for metropolitan area Ethernet. PBT is based on using existing IEEE 802.1 VLAN tags to deliver Ethernet services across a provider network. PBT competes head-to-head with T-MPLS, and the jury's still out on which one will gain the most traction.

Finally, a variant of MPLS called Generalized Multiprotocol Label Switching (GMPLS) gives routers the ability intelligently signal the optical layer, enabling providers to establish, change or tear down optical links in real time. Thus, service providers can provision "optical wavelength" services based on MPLS.

MPLS (Multi Protocol Label Switching)

MPLS has its roots in Ipsilon's IP Switching, Cisco's Tag Switching, IBM's ARIS technology and a few other proposals to bring the sort of traffic engineering found in connection-oriented Asynchronous Transfer Mode and frame relay networks to connectionless IP networks.

The idea is to steer IP traffic onto a variety of routes instead of the single one discovered by an interior gateway protocol such as Border Gateway Protocol, to avoid congestion or failures, or to enable a particular class of service or guaranteed service level.

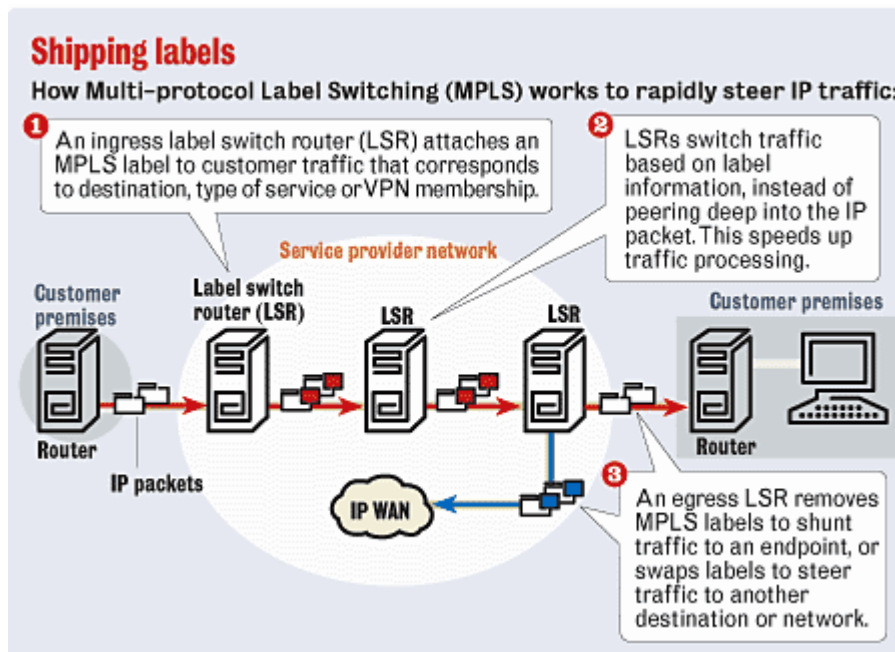
MPLS switches and routers affix labels to packets based on their destination, type-of-service parameters, Virtual Private Network membership or other criteria. As a packet traverses a network, other switches and routers build tables associating packets and routes with labels.

GUIDE TO MPLS

By J Doyle

The MPLS switches and routers - dubbed label switch routers - assign each packet a label that corresponds to a particular path through the network.

All packets with the same label use the same path - a so-called label switched path (LSP). Because labels refer to paths and not endpoints, packets destined for the same endpoint can use a variety of LSPs to get there.



Understanding MPLS Label Stacking

Label stacking is the encapsulation of an MPLS packet inside another MPLS packet – that is, adding an MPLS header “on top of” (hence stacking) an existing MPLS header. The result of stacking is the ability to tunnel one MPLS LSP inside another LSP.

Figure 1 shows an example of label stacking. LSP 1 is tunneled inside LSP 2; the ingress, transit, and egress routers for LSP 2 are shown. At the ingress router, an MPLS packet (belonging to LSP 1) arrives with a label of 22. Rather than the SWAP that an MPLS switching table would normally perform, in this case the table specifies a PUSH: An MPLS encapsulation. So a header is added with a label of 18 and the packet is forwarded out interface 1 to the next router.

Notice that across LSP 2 all the switching is performed on the outer label; the inner label is never observed or changed until the outer header is removed and the packet egresses LSP 2.

GUIDE TO MPLS

By J Doyle

So why is label stacking important? One reason is in order to take advantage of the positive aspects of both of the LSP signaling protocols, Label Distribution Protocol (LDP) and Resource Reservation Protocol with Traffic Engineering Extensions (RSVP-TE).

The primary advantage of LDP is that it scales well. It signals LSPs hop-by-hop, and so routers along the path do not have to maintain state for each LSP. Therefore LDP is useful in edge applications such as VPNs where hundreds or thousands of LSPs are originated and terminated. But LDP has no traffic engineering capabilities; it just follows the IGP shortest path to find LSP end-points.

RSVP-TE, on the other hand, supports traffic engineering, enabling efficient infrastructure resource utilization and some very useful link and node protection features (perhaps the subject of a future post). You can specify a number of constraints on the path an LSP takes, diverging from the IGP shortest path. To do this, RSVP-TE signals LSPs end-to-end; and as the name implies, it reserves infrastructure resources. And those reservations mean that the routers must keep state for each of the LSPs it maintains (ingress, transit, or egress), and thus RSVP-TE does not scale as well as LDP.

Large-scale MPLS networks can take advantage of the best characteristics of both of these protocols by using LDP for the large numbers of service- and customer-specific LSPs originating and terminating at the network edge, using RSVP-TE to create traffic-engineering LSPs between PoPs, and then tunneling the LDP-signaled edge LSPs inside of the RSVP-signaled core LSPs.

Another application for label stacking is in the creation of MPLS VPNs. That's really the point of covering stacking in this post, and the subject of the next couple of posts. Stay tuned.

Understanding the Role of FECs in MPLS

It's funny how things come in waves. For most of last year the majority of my consulting engagements concerned IPv6 in some way or another. But over the past month and a half most of my time has been focused on conducting MPLS seminars of various sorts and for varied audiences.

A central concept to MPLS is the Forwarding Equivalence Class (FEC), and it's something many people new to the technology struggle to understand. So in this post I'd like to discuss FECs and their role in MPLS.

An FEC is a set of packets that a single router:

- (1) Forwards to the same next hop;
- (2) Out the same interface; and
- (3) With the same treatment (such as queuing).

GUIDE TO MPLS

By J Doyle

FECs are nothing new. Every router performing generic IP forwarding determines the next hop to which the packet is to be forwarded, the interface out which the packet is sent to get to that next hop, and how to queue the packet for that interface. But we don't often hear those very basic procedures presented as "determining what FEC a packet belongs to." On the other hand, FECs are almost always discussed when introducing the fundamental concepts of MPLS. The reason for this is that understanding how a packet's FEC is determined at an MPLS Label Switching Router (LSR) goes a long way toward understanding MPLS itself.

First let's look at how a packet is forwarded across a path toward a destination using regular IP processes. Figure 1 shows four routers. A packet arrives at R1, and its destination IP address is examined. A lookup is performed, the packet's FEC (again: next-hop, outgoing interface, and forwarding treatment) is determined, and using that information the packet is forwarded to the next hop router R2. R2 then repeats the process: The FEC is determined and the packet is forwarded to R3. R3 again determines the packet's FEC and forwards it to

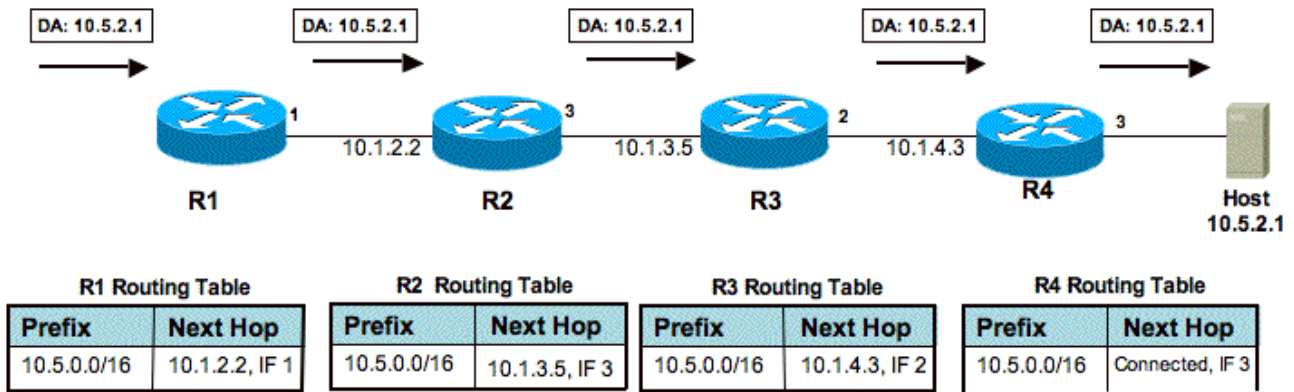


Figure 1
In normal IP routing, the packet's FEC is determined at each hop along the path to the destination.

GUIDE TO MPLS

By J Doyle

R4.

In other words, the packet's FEC is determined hop-by-hop at every router along the forwarding path toward the destination. Now let's suppose the four routers of Figure 1 are MPLS LSRs, and there is a Label Switching Path (an LSP, which is simply an MPLS virtual

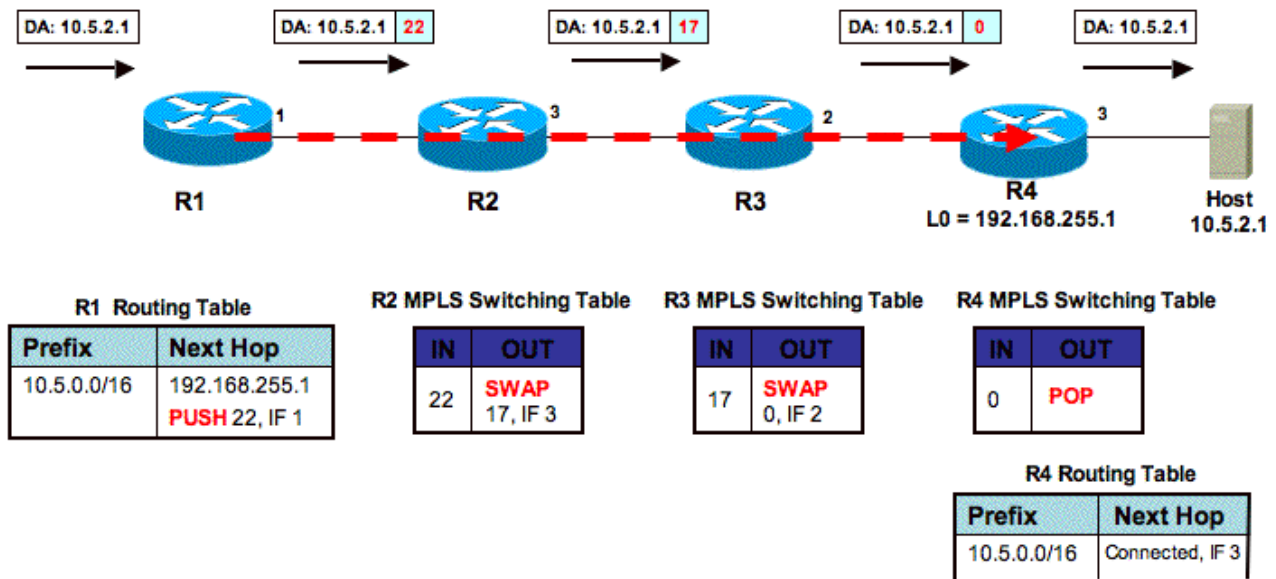


Figure 2

The packet's FEC is determined only at the ingress to the MPLS LSP, and is not determined again until the packet exits the LSP.

circuit) between R1 and R4; the termination point of the LSP is R4's loopback interface, 192.168.255.1. This relationship is shown in Figure 2.

As before, a packet arrives at R1 and the packet's FEC is determined. In this case, however, the next hop for the FEC is the termination point of the LSP at R4. The packet is encapsulated with an MPLS header – that is, an MPLS header is PUSHed onto the packet – and the packet is forwarded out the interface to R2. At the remaining hops to R4, the packet is simply switched from its incoming interface to the outgoing interface designated by the

GUIDE TO MPLS

By J Doyle

MPLS switching table, with the MPLS label being SWAPed at each hop. At no point along the LSP does a router again determine the packet's FEC until the packet exits the LSP (the MPLS header is POPed) at R4.

And that's the key point: In an MPLS network the FEC is determined only once, at the ingress to an LSP, rather than at every router hop along the path. In our example, even though the LSP actually traverses R2 and R3, R1 "sees" the LSP as a single link to R4 and therefore chooses it as a better path than the hop-by-hop IP routed path through R2 and R3.

Conceptually, then, you can think of MPLS as a technology that pushes the "intelligence" to the edge of the network, leaving the core to do simple switching. In other words, as Figure 3 illustrates, the network control plane is located at the edge and the forwarding plane is in the center. This separation of control and forwarding planes is very much in keeping with broad trends in networking, such as in high-performance routers that have long implemented the control and forwarding planes as separate physical components.

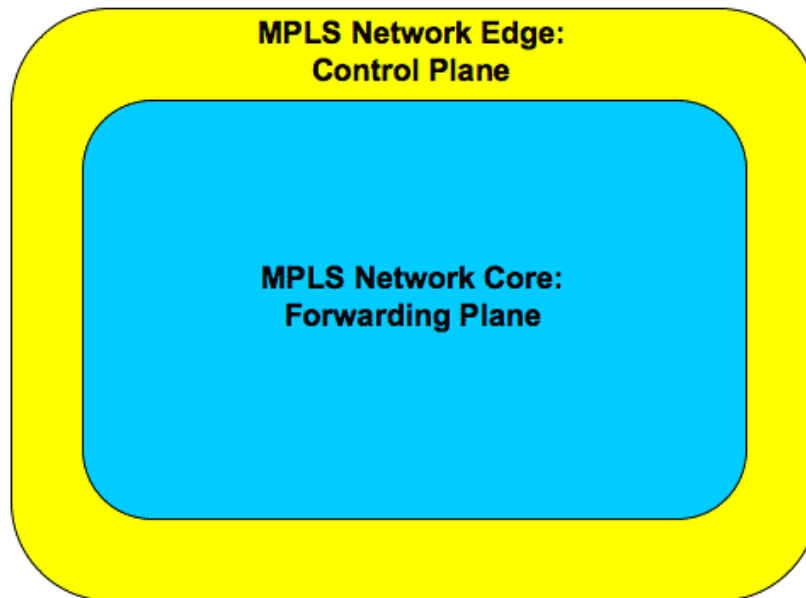


Figure 3

Conceptually, the single FEC determination at the LSP ingress means that the MPLS network control plane is at the edge and the MPLS network core is the forwarding plane.

GUIDE TO MPLS

By J Doyle

Understanding MPLS Explicit and Implicit Null Labels

Discussing the role of the FEC in MPLS networks, I used in one of my examples a particular label value; I was hoping someone would ask about it, which would give me a nice segue to this post. Well no one did, but I'm going to tell you about it anyway. The label, used between the next-to-last LSR (called the penultimate LSR) and LSR on which an LSP terminates (called the egress LSR), is called the IPv4 Explicit Null label, and has value of 0. (There is also an IPv6 Explicit Null label, with a value of 2.)

Figure 1 shows the use of the explicit null label. When an LSR receives an MPLS header in which the label is set to 0, it always POPs the header – that is, it removes the label. The use of the explicit null is in keeping with the concept of a packet's FEC being determined only at the ingress to an LSP, and no where else along the path. The reserved label value of 0 prevents the egress LSR from either having to keep some sort of state designating a set of its allocated labels as POP labels, or having to look to the encapsulated payload to determine what LSP it belongs to.

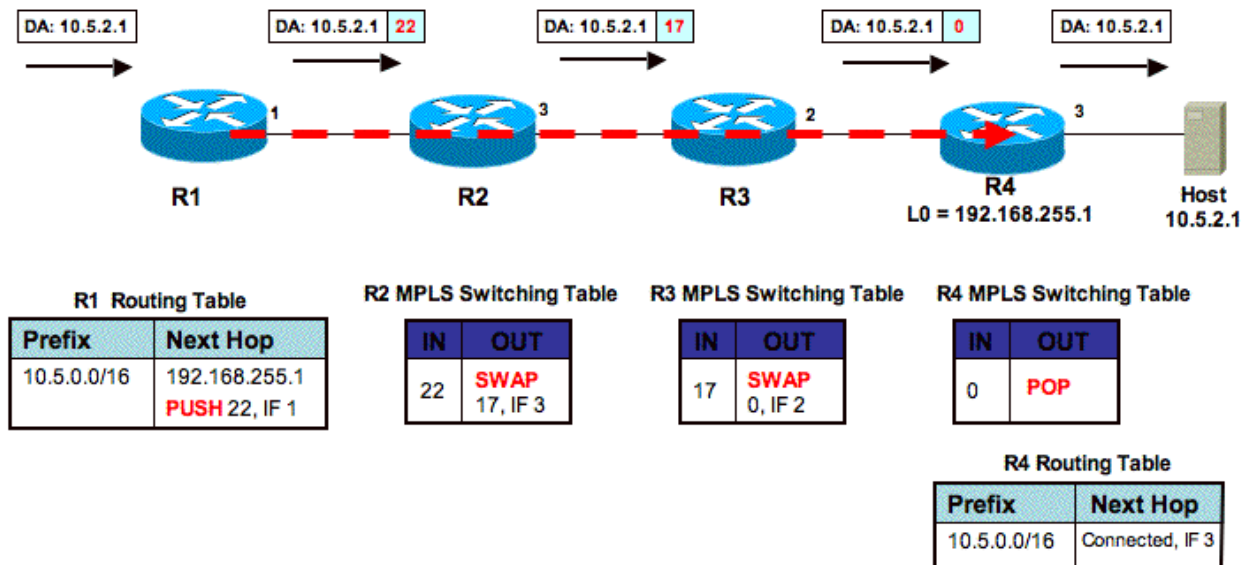


Figure 1

The IPv4 Explicit Null label, label 0, tells an LSR to POP, or remove, the MPLS header.

GUIDE TO MPLS

By J Doyle

There's something to notice about the switching and forwarding procedure depicted in Figure 1: Once R3, the penultimate LSR, has switched the packet out its interface IF2 toward R4, the label has no further relevance except to tell R4 to POP the header; that is, no further MPLS switching is performed. So in fact once R3 has determined what interface to switch the packet out, it could go ahead and POP the label there and save a step at R4.

This procedure – with the wonderful name Penultimate Hop Popping – is depicted in Figure 2. R3's MPLS switching table indicates, for the incoming label of 17, an outgoing label of 3. But then this label value – called the Implicit Null – resides in the table as an outgoing label, an LSR does not perform a SWAP; instead it performs a POP and forwards the packet out the referenced interface (in this example, R3's IF2).

The value of penultimate hop popping is that it saves a small step at the egress router of having to first consult its MPLS switching table, POP the header, and then examine the decapsulated payload to determine the proper next step. Instead, in Figure 2, R4 just sees an incoming IP packet and takes the appropriate action for generic IP routing. The question arises, then: If implicit null is more efficient, why do we need explicit null at all? The answer is Class of Service.

When a packet or Ethernet frame is encapsulated in MPLS, you have the option of copying the IP precedence or 802.1p bits to the three CoS bits of the MPLS header (unfortunately called the Experimental or EXP bits), so that the MPLS LSP provides the same CoS behavior, or you can set the EXP bits independently, so that the LSP CoS behavior has a designated CoS behavior that is independent of any encapsulated payload.

In this second case, you would want to use the explicit null label between the penultimate and egress LSRs. If a POP is performed at the penultimate LSR, as in Figure 2, the EXP bits in the MPLS header are no longer available as a reference for queuing and the packet is queued on the outgoing interface according to the CoS behavior of the underlying payload. An explicit null, on the other hand, leaves the MPLS header in place until it reaches the egress, preserving the LSP CoS behavior across the entire LSP.