The GMPLS Control Plane Architecture for Optical Networks: Routing and Signaling for Agile High Speed Switching

David Griffith

Abstract

Optical networking technology underwent a major revolution in the 1990s as the old paradigm of Synchronous Optical Network/Synchronous Digital Hierarchy (SONET/SDH) systems to support circuit-switched connections began to give way to a new mesh network of transparent, high-speed optical switches with the capability to support a variety of higher-layer traffic types and services. In order for such a network to operate efficiently, it is essential that it employ a control plane that is capable of managing both legacy framed traffic and new traffic types. Also, the optical control plane must be able to operate across network boundaries, both at the edge interface to the customer's equipment and at administrative domain boundaries in the core network. The Internet Engineering Task Force (IETF) has tasked several working groups to develop the architecture for such a control plane as well as protocols to support its functioning. These groups' work has built on previous work in the IETF on Multi-Protocol Label Switching (MPLS), which was developed to allow packet routers to operate more efficiently. In this chapter, we describe the Generalized MPLS (GMPLS) architecture and related protocols, specifically the Resource reSerVation Protocol with Traffic Engineering (RSVP-TE) for signaling, Open Shortest Path First with Traffic Engineering (OSPF-TE) for routing, and the Link Management Protocol (LMP).

Index Terms

GMPLS, WDM, CR-LDP, RSVP-TE, ISIS-TE, OSPF-TE, LMP, protection and restoration, traffic engineering

I. Introduction

Optical communications technology has experienced tremendous growth in the nearly 30 years since the first telephone calls were transmitted commercially over fiber by AT&T. Within a few decades, improvements in semiconductor lasers, optical detectors, tunable gratings, transparent optical switch fabrics, optical fibers, and other physical layer technologies have produced a vast network that can transmit information on multiple wavelengths at rates of up to 40 Gbps on each wavelength. Much of this expansion was created by a burst of demand during the 1990s that was caused by the public's enthusiastic adoption of computer networking services in the wake of the introduction of the World Wide Web.

Optical networks were originally conceived as high-bandwidth systems that could multiplex large numbers of voice calls while meeting the strict grade-of-service requirements demanded by the traffic that they were carrying. The steady growth of data traffic, which now constitutes the majority of global information flow, has produced a need for a network control plane that is capable of supporting both circuit-switched and packet-switched connections. In addition, the steady rollout of services that require ever-greater bandwidth, such as video on demand (VOD), is generating a need on the part of major carriers for the ability to create, modify, and tear down connections rapidly and with a minimum amount of human intervention.

The first set of optical standards, which emerged in the early 1990s, described mechanisms to multiplex TDM digital signals using hierarchical framing. Two similar sets of standards were developed: a version used in North America, known as Synchronous Optical Network (SONET), the other version, used virtually everywhere else in the world, is called Synchronous Digital Hierarchy (SDH). Optical communications systems using SONET/SDH were widely deployed during the last decade but they were intended primarily to support circuit-switched voice traffic. As the quantity of packet-switched traffic has increased, the SONET/SDH optical networking model has come to be viewed as adding too much complexity and expense to data network implementations. It is more efficient to run Internet Protocol (IP) traffic directly over optics, eliminating the intervening layers, but this requires an optical control plane that can support all of the administrative functions that such a network requires.

For the past several years, the Internet Engineering Task Force (IETF) and other standards development organizations (SDOs) have been working to define the architecture and protocols that can be used to compose a flexible control plane for optical networks. The IETF's work has built on the development of MultiProtocol Label Switching (MPLS), extensions to existing routing and signaling protocols, and the creation of new protocols where needed. The resulting generalized MPLS (GMPLS) architecture offers the possibility of efficient, automated control of optical networks supporting a variety of services. This chapter examines the structure of the GMPLS control plane and describes the basic functions of its parts.

The author is with the National Institute of Standards and Technology (NIST), 100 Bureau Drive, Stop 8920, Gaithersburg, MD 20879
email: david.griffith@nist.gov

This chapter is organized as follows. In Section II, we describe the essential features of the GMPLS architecture. We discuss how the label switching paradigm was extended to encompass a large range of interface types, and the types of optical networking architectures that are possible under GMPLS. In Section III, we discuss the signaling protocols that are used for connection establishment and maintenance, particularly RSVP-TE. In Section IV, we examine routing in GMPLS networks. We focus on OSPF-TE, which is an extension of link-state routing for packet-switched networks to support constrained routing in networks that contain a variety of interface types and whose elements do not necessarily share traditional types of routing adjacencies. In Section V, we discuss the Link Management Protocol (LMP), which was developed to support the creation of control channels between nodes and to verify interface mappings between connected switches. We conclude in Section VI by comparing the work on GMPLS in the IETF to related work taking place in other SDOs.

II. The GMPLS Architecture

Generalized Multiprotocol Label Switching (GMPLS) [1], [2] grew out of the development of MPLS and was prompted by the need for an IP-oriented control plane that could incorporate traffic engineering capabilities along with support for a multitude of traffic types, particularly legacy systems such as ATM and SONET/SDH. GMPLS builds on the label switching model while adding features that enable it to work with optical networks.

A. MultiProtocol Label Switching (MPLS)

MPLS evolved from several parallel development efforts. The story of its creation has been recorded in detail elsewhere [3], but we give the main details here. In the early 1990s, Asynchronous Transfer Mode (ATM) was a strong contender for dominance in large data networks, mostly because ATM switches at that time were faster than IP routers. Because this situation created a need to tunnel IP traffic over ATM networks, the IETF developed a set of Requests for Comments (RFCs), which are the set of freely available documents that define Internet protocols, to describe how this could be done. Several router vendors began working on an IP/ATM architecture that did not use ATM's User-Network Interface (UNI) signaling (described in the ATM Forum's UNI 1.0 document [4] and by the ITU [5]), which was considered cumbersome, in favor of a more lightweight approach. The need for a standardized version of label-based switching motivated the creation of a working group in the IETF.

The version of label switching that ultimately emerged from the MPLS working group incorporates elements of all the major vendor schemes but is most closely related to the approach, proposed by engineers from Cisco, known as tag switching. Tag switching incorporated mechanisms for forwarding IP datagrams over multiple link layer protocols such as ATM but also the Point-to-Point Protocol (PPP) and IEEE 802.3. Tag switching was novel in that it used the RSVP protcol to set up connections with specific traffic characteristics in order to support some degree of Quality of Service (QoS).

The idea behind all the approaches that the IETF harmonized to create MPLS is that of labels that can be applied to IP packets to forward them without having to examine the IP header's contents. The labels are encoded using a thin encapsulation header (known as a shim header) to tunnel IP flows through various types of layer 2 clouds [6]. The shim header is located between the IP header and the link layer protocol header.

MPLS uses forwarding adjacencies (FAs) to set up label switched paths (LSPs) across networks of label switching routers (LSRs). All IP packets that have some set of parameters in common are considered to belong to a particular FA. In practice, membership in a FA is based solely on the IP packet's destination address field. Once a complete set of label mappings has been installed in each LSR on the LSP for an FA, packets belonging to that FA can be forwarded over the LSP from the ingress LSR to the egress LSR without having the contents of their IP headers examined or modified. For this reason, the MPLS architecture describes mechanisms that can be used to decrement the IP header's Time To Live (TTL) field by the appropriate amount when the packet leaves the MPLS cloud [7].

MPLS supports hierarchical LSPs by allowing multiple shim headers to be stacked between the layer 2 and layer 3 headers, so that LSPs can be tunneled through other LSPs. The label stack is processed using the last-in-first-out (LIFO) discipline, so that the label associated with the highest LSP in the hierarchy is located at the bottom of the stack. When a packet bearing a label stack arrives at the input port of an intermediate LSR (i.e., an LSR that lies along the packet's path but is neither the source nor the destination), the LSR removes (or "pops") the topmost label from the stack and uses it to locate the appropriate Next Hop Label Forwarding Entry (NHLFE), which indicates to which output port the packet should be sent and which value should be assigned to the outgoing label that will be applied to the packet. A new label with the appropriate value is then created and it is pushed onto the top of the stack before the packet is forwarded to the next LSR.

An illustration of how labels are used to forward packets is given in Figure 1. The router shown in the figure is an intermediate LSR for the LSP that is carrying Packet A. The LSR examines Packet A's label and uses the NHLFE table to determine the output port to which it should send Packet A and the outgoing label that it should apply to the departing packet. Packet B is being tunneled through a lower-layer LSP, and the LSR in the figure is the egress point for that LSP. In this case, the LSR removes the topmost label and forwards the packet, whose remaining label is meaningful to the LSR that is directly downstream (i.e., towards the destination) from the LSR in the figure.

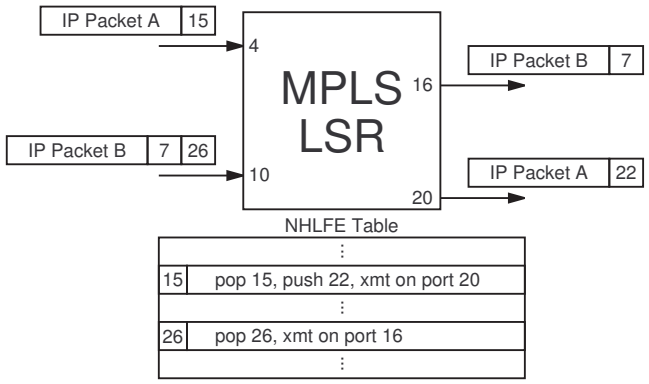


Fig. 1. An example of packet forwarding in an MPLS router.

B. Generalizing MPLS

As the development of the MPLS architecture proceeded, people realized that there were several interesting parallels between the tunneling of IP traffic along LSPs consisting of a sequence of LSRs and the proposed tunneling of IP traffic along lightpaths consisting of a sequence of optical cross-connects (OXCs) with wavelength conversion capabilities. If an optical network could be made to support a kind of MPLS architecture, wavelengths could play the same role as numerical labels in an electrical MPLS network of routers. Instead of a table of next hop label forwarding entries, each OXC would maintain a table of port and wavelength mappings, so that data associated with the lightpath arriving on any particular input port on a particular wavelength would automatically be mapped to a predetermined wavelength on a desired output port. The set of port and wavelength mappings would be implemented by configuring the switching fabric in the switch's core.

Using this basic realization, the IETF began work in early 2000 on the GMPLS architecture [1]. Originally known as MPλS or MPLambdaS, it is an extension of the MPLS architecture that includes mechanisms for encoding labels that are associated with interfaces that are not packet-switch capable. The GMPLS architecture supports the following five interface types.

1) Packet Switch Capable (PSC)
   An interface that is uses the information contained in the packet header to forward individual packets.
2) Layer-2 Switch Capable (L2SC)
   An interface that can read layer 2 frame headers and use them to delineate individual frames and forward them.
3) Time-Division Multiplex Capable (TDM)
   An interface that switches data based on the position of the slot that it occupies in a TDM frame.
4) Lambda Switch Capable (LSC)
   An interface that switches traffic carried on an incoming wavelength onto a different outgoing wavelength.
5) Fiber Switch Capable (FSC)
   An interface that switches information streams or groups of streams based on the physical resource that they occupy.

GMPLS is supported by separate protocols that perform routing (OSPF-TE and ISIS-TE), signaling (RSVP-TE and CR-LDP), and link management and correlation (LMP). Their places in the IP-based protocol stack are shown in Figure 2. Each protocol name is connected to the layer that it uses to encapsulate and transport its messages. In only two cases (LMP and CR-LDP) are layer 4 segments used to carry GMPLS messages. The stack shown in the figure is a "heavy stack," in that a large number of layers reside between the IP layer and the optical layer. Future architectures use protocol stacks with fewer layers, or even let IP traffic run directly over optics, without any framing technologies in between. While MPLS supports a variety of label distribution modes, GMPLS protocols work solely with the downstream-on-demand mode, in which label mappings are distributed by LSRs to their upstream neighbors only in response to a label mapping request that comes downstream. This enables GMPLS to support circuit-switching operations. Labels in GMPLS, known as generalized labels, are carried in signaling messages and are typically encoded in fields whose internal structure depends on the type of link that is being used to support the LSP. For example, Port and Wavelength labels are jointly encoded in a 32-bit field and have meaning only to the two nodes exchanging label mapping information. MPLS shim labels and Frame Relay labels are encoded right-justified in the 32-bit general label space. For ATM labels, the Virtual Path Identifiers (VPIs) are encoded right-justified in bits 0-15, and the Virtual Channel Identifiers (VCIs) are encoded right-justified in bits 16-31. Encodings also exist for SONET/SDH time slots, as shown in Figure 3. They are as follows:

III. GMPLS Signaling

The MPLS architecture does not state explicitly how labels are to be requested and distributed. The mechanism for doing this is left up to the network operator, although the IETF has created a signaling architecture for GMPLS [8]. Tag switching, the

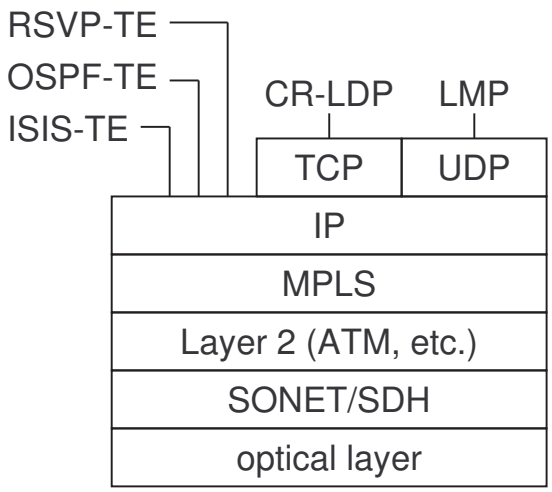
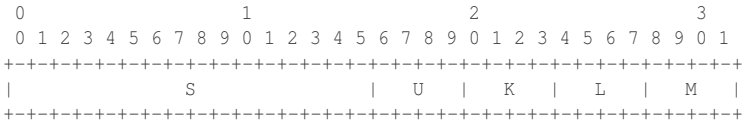


Fig. 2. The GMPLS protocol stack, showing relative positioning of essential routing, signaling, and link management protocols.



S: Index of a particular STS-1 SPE in a STS-N multiplex.
U: (SDH only) Index of a particular virtual container (analogous to virtual tributary) in an STM-1
K: (SDH only) Index of a particular branch in a VC-4.
L: Index of a branch in a TUG-3, VC-3, or STS-1 SPE. If L = 1 the SPE is not subdivided; L = 2,3,…,8 indicates one of 7 possible VT groups in the SPE (if SONET is used).
M: Index of a branch in a VT group. M = 0 indicates an unstructured STS-1 SPE. Otherwise it refers to a particular VT-1.5, VT-2, or VT-3.

Fig. 3. Encodings for SONET/SDH indices in the generalized label.

MPLS ancestor developed by Cisco, used RSVP to request and distribute labels. It was therefore natural that RSVP would be proposed as the standard protocol for signaling LSP setup and teardown in GMPLS. A small group of equipment vendors led by Nortel networks proposed a competing signaling scheme, known as the Label Distribution Protocol (LDP), which they had created explicitly for MPLS. LDP was later extended to support constrained routing (CR) [9]. CR-LDP's proponents pointed out several shortcomings of conventional RSVP such as its lack of scalability due to its need to periodically refresh connection state information by retransmitting signaling messages. This issue and others were addressed in the modified version of RSVP described in RFC 3209 [10]. Additional modifications to RSVP, specifically to support traffic engineering and GMPLS, are described in RFC 3473 [11].

The struggle for dominance between the two protocols lasted for several years and affected discussions in other SDOs, such as the OIF, as well as those in the MPLS working group and other working groups in the sub-IP area in the IETF. For approximately two years, the issue was addressed by "agreeing to disagree" and supporting both proposed signaling protocols. This led to considerable document bloat in the affected working groups as each proposed signaling protocol extension required defining new objects for RSVP-TE and for CR-LDP. The issue was resolved after the IETF's meeting in July, 2002, when RSVP-TE was designated as standards track, while CR-LDP was relegated to informational status [12]. This decision was influenced by market realities, which were reflected in an implementation survey conducted by the MPLS working group [13] that showed that while RSVP-TE was supported by nearly all of the respondents, CR-LDP was implemented by a small minority, and those vendors who supported CR-LDP also tended to support RSVP-TE. For this reason, in this section we describe only RSVP-TE and how it is used to manage connections in optical networks.

A. RSVP-TE

The RSVP protocol and its extension, which are described in detail in [10], [11], and [14], was originally designed to support Integrated Services (IntServ) in IP networks. It does this by reserving resources in routers to achieve a desired QoS. RSVP

is concerned only with signaling, and it is functionally separate from other networking functions such as routing, admission control, or policy control.

1) RSVP Operations: RSVP in its classical form supports receiver-initiated reservations for multicast sessions. Applications with data to transmit, known as senders, advertise their status by transmitting Path messages downstream to one or more receivers. As Path messages traverse the network, they establish state information in the RSVP-capable routers that they pass through. This information generally consists of a traffic specification that includes information necessary to support QoS functions (e.g., peak data rate, peak burst size, etc.). It also contains information that identifies the sender that created the Path message and the router immediately upstream (i.e., toward the sender) from the router that received the message. Once a Path message reaches its destination, that node can start sending Reservation (Resv) messages upstream to the originating source node. As the Resv message propagates toward the sender, it causes RSVP-capable routers along the route to reserve resources to support the traffic characteristics (or flowspec) that are advertised in the Resv message. When the session's sender receives a Resv message, it can begin sending data to the receiver. Because the exchange of Path and Resv messages supports only unidirectional flows, a separate set of Path and Resv messages must be exchanged to support a bidirectional session. The structure of Path and Resv messages in RSVP-TE for GMPLS is shown in Figure 4 and Figure 5, respectively. In both figures, gray fields indicate optional objects. In Figure 5, the structure of the flow descriptor list depends on the filter being used by the source that generated the Path message. A list can include any number of label blocks, but only one label can be associated with each FILTER_SPEC in the list.

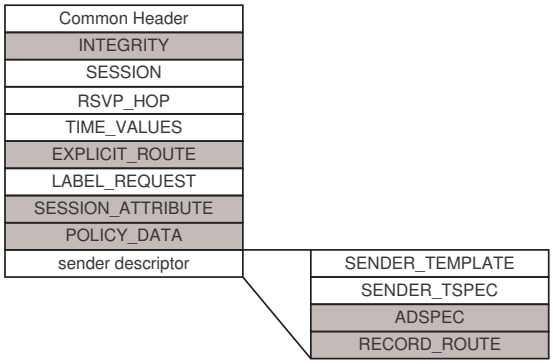


Fig. 4. Structure of the Path message in RSVP-TE.

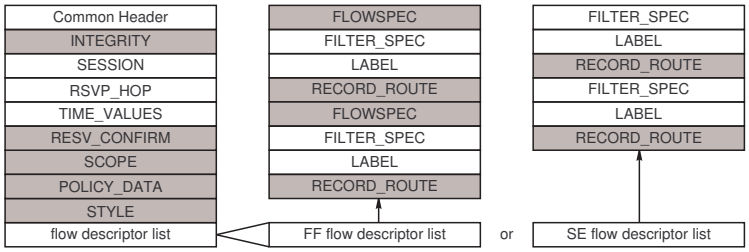


Fig. 5. Structure of the Resv message in RSVP-TE.

When the Multiprotocol Label Switching protocol was being developed by the MPLS working group, RSVP was extended to allow it to support traffic engineering (TE) by requesting and distributing label bindings [11]. They are used to support the creation of LSP tunnels, i.e., LSPs that are used to tunnel below standard IP-based routing. The modified version of RSVP, known as RSVP-TE, builds support for MPLS into RSVP by defining new objects for transporting label requests and mappings using Path and Resv messages. Details about the Forwarding Equivalence Class (FEC) that is to receive the mapping are encoded in new versions of the SESSION, SENDER_TEMPLATE, and FILTER_SPEC objects that identify the ingress of the LSP tunnel. In addition, LABEL_REQUEST and LABEL objects have been added to the Path and Resv messages, respectively. RSVP-TE supports only downstream-on-demand label distribution mode with ordered control. The LABEL object allows the Resv message to carry a label stack of arbitrary depth. The LABEL_REQUEST and LABEL objects must be stored in the Path and Reservation state blocks, respectively, of each node in the LSP tunnel, and they must be used in refresh messages, even if there has been no change in the tunnel's state. This will tend to increase the signaling overhead. If an intermediate node doesn't support LABEL_REQUEST objects or has no label bindings available (e.g., all its resources are in use), it sends a PathErr message back to the source node.

If traffic engineering is being used to route LSP tunnels, there are a number of situations where an active tunnel can be rerouted, such as when there is a change of next hop at a particular point in the tunnel or when a node or link failure occurs. When tunnels are rerouted, the preferred course of action is to set up the alternate route before tearing down the existing route so that there is no disruption of the data traffic (this process is known as "make before break"). In order to support this functionality, RSVP-TE must use the shared explicit (SE) reservation style, in which an explicitly-defined set of senders is associated with a given session. This allows the tunnel ingress to specify a tunnel detour associated with a new label descriptor at the same time that it maintains the old tunnel, over which data is still flowing. After the detour has been established, the defunct portion of the tunnel can be torn down or allowed to time out.

Once a receiver is finished receiving data or a sender no longer has data to send, they can delete the state that they created by respectively using ResvTear and PathTear teardown messages. Originally RSVP used soft state, so that it was possible to avoid using teardown messages and simply allow the state to time out on its own; however, this is discouraged in [14] and circuit-switched applications require explicit teardown messages. Intermediate routers along a path can also send teardown messages if either the Path or the Reservation state associated with a particular session times out. The messages are forwarded immediately by each router that receives them until they reach a node where they do not cause that node's state to change. Because these messages do not run over a transport layer protocol that can request retransmission of missed segments, they can be lost. In old RSVP networks this was not a fatal problem because the state would time out on its own eventually; when this happened the node whose state had timed out would transmit the appropriate teardown messages and the ordered teardown process would continue. RSVP-TE uses message acknowledgement to prevent the loss of teardown messages.

2) Explicit Routing and Route Recording: RSVP-TE also includes support for explicit routing by incorporating an EXPLICIT_ROUTE object (ERO) into the Path message and a RECORD_ROUTE object (RRO) into both the Path and the Resv message. These objects cannot be used in mulitcast sessions. The RECORD_ROUTE object is used to do loop detection, to collect path information and report it to both ends of the tunnel, or to report the path to the tunnel ingress so that it can send an EXPLICIT_ROUTE object in its next Path refresh to pin the route. The EXPLICIT_ROUTE object contains a sequence of abstract nodes through which the LSP tunnel must pass. The ERO can be modified by nodes that forward it. RSVP-TE can also specify nodes that should be avoided [15].

3) RSVP-TE Hello Messages: Unlike RSVP, RSVP-TE allows directly connected neighbors to exchange Hello messages to detect node failures. The Hello feature is optional. Hello messages carry either HELLO_REQUEST or HELLO_ACK objects, both of which contain 32 bit instance numbers for the nodes at both ends of the connection. A node that uses the Hello option sends Hello_Request messages to its neighbor at regular intervals (the default interval is 5 msec); a participating recipient replies with a Hello_Ack message. If the sender of the Hello_Request hears nothing from the receiver after a fixed period of time (usually 3.5 Hello intervals), it assumes that the link between them is broken. If either node resets or experiences a failover, it uses a new instance number in the Hello messages it transmits. This allows an RSVP-TE node to indirectly alert its neighbor that it has reset. If there are multiple numbered links between neighboring nodes (i.e., each interface has its own IP address), then Hello messages must be exchanged on all the links.

4) Label Suggestion and Upstream Labels: Optical lightpaths are typically bidirectional and follow the same route. While it is possible to create a bidirectional lightpath that comprises two disjoint unidirectional paths, this creates management problems for the network operator and is avoided whenever possible. Thus, when a bidirectional lightpath is created, there will be two exchanges of signaling information, one for each unidirectional component.

For MPLS networks operating in downstream-on-demand mode, labels are not installed in a LSR until a Resv message is received. This mode of operation poses problems when it is applied to photonic switches that route light directly from input ports to output ports without converting the received signal into electrical form. Switch fabrics based on Micro Electro-Mechanical Systems (MEMS) (e.g., tiltable mirrors) require tens of milliseconds to respond to reconfiguration commands. This adds considerably to the LSP establishment time.

To reduce the time required to set up a new connection, RSVP-TE can include suggested labels in Path messages. The suggested label allows the upstream OXC to tell its downstream neighbor from which interface it would like to receive traffic. When an OXC transmits a Path message bearing a suggested label, it begins configuring its own switch fabric in anticipation of being able to receive data on the suggested interface. When the OXC receives a Resv message from its downstream neighbor, it checks the label contained in the message. If this label matches the suggested label, the OXC continues configuring itself; otherwise it aborts the configuration changes associated with the suggested label and begins configuring its switch fabric in accordance with the label mapping contained in the Resv message.

To further expedite the lightpath creation process, OXCs can include an upstream label object in the Path message. This object informs the downstream neighbor which interface it should use for the reverse direction counterpart to the lightpath that is being actively signaled. An example of label suggestion and upstream labels appears in Figure 6. In the figure, each ordered pair (p, λ) denotes a particular wavelength λ on a particular fiber port p. OXC A begins by sending a Path message to OXC B that suggests transmitting on λ_2 on OXC B's port 1 for the upstream path and assigns λ_1 on OXC's port 6 for the upstream path. OXC A begins configuring its switch fabric while OXC B sends a Path message to OXC C that assigns λ_1 on port 7 of OXC B for the upstream path and suggests λ_2 on OXC C's port 1 for the downstream path. OXC C is unable to honor the suggestion and instead assigns λ_3 on its port 2, forcing OXC B to restart its configuration process. OXC B confirms OXC A's

suggestion in a Resv message, and both lightpaths are established.

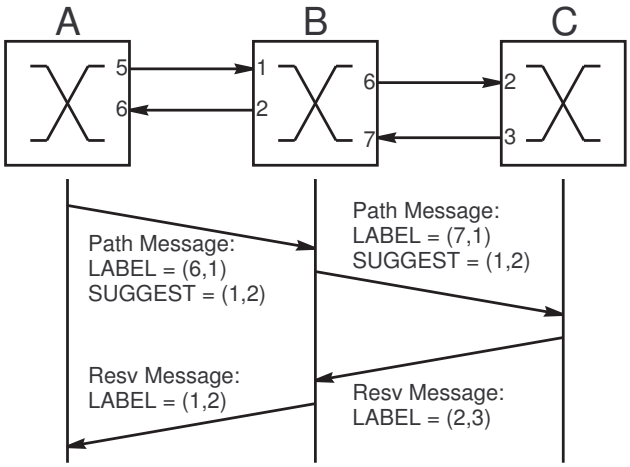


Fig. 6. Label suggestion and upstream labels for path establishment.

An important issue associated with lightpath setup is the problem of contention for resources by multiple LSPs. For an example of this consider Figure 6, and suppose that OXC B sends a Path message to OXC A suggesting that λ_2 on port 1 be used for the reverse path, which collides with OXC A's upstream label assignment. GMPLS uses a simple mechanism to resolve contention issues; the node whose node_ID is greater wins the contention. Assuming OXC A's node_ID is greater, OXC B is forced to use a different label.

IV. GMPLS Routing

One of the most important functions of an optical network management system is determining which resources should be assigned to support a new traffic flow. GMPLS uses IP routing with extensions for traffic engineering to carry out this function, using constrained routing to account for restrictions that the physical layer or network operator may impose (e.g., routing diversity for paths that share a common protection resource). Both the Open Shortest Path First (OSPF) [16] and Intermediate System-to-Intermediate System (IS-IS) routing protocols were extended to support GMPLS routing features [17]; we focus on OSPF-TE in this section.

A. OSPF-TE

The OSPF-TE extensions use the concept of opaque Link State Advertisements (LSAs), which are defined in RFC 2370 [18]. Three types are defined, and in each case the Type Value field determines the flooding scope. Opaque LSAs with a Type Value of 9 are flooded only on a local link or subnet. Opaque LSAs whose Type Value is 10 are flooded within a single area in an Autonomous System (AS), while those whose Type Value is 11 are flooded throughout an entire AS. Because the Type Value field is used to indicate flooding scope, the Link State ID field is used to differentiate the various types of opaque LSAs. The first eight bits of the Link State ID contain the Opaque Type field, while the other 24 bits contain the Opaque ID, which serves the same function as the Link State ID Field in non-opaque LSAs. Opaque LSAs are flooded only to routers that are capable of understanding them; each router's knowledge of the ability of its neighbors to understand opaque LSAs is created during the database exchange process.

OSPF-TE [19] includes a new Traffic Engineering LSA for carrying attributes of routers (and switches), point-to-point links, and connections to multiaccess networks. The TE LSA has an LS Type value of 10, so it is flooded only within areas, not over an entire AS. Network LSAs (Type 2 LSAs) are also used in TE routing calculations. Instead of the Link State ID that typically appears in the LSA header, the TE LSA uses an LSA ID that consists of 8 bits of type information and a 24-bit Instance field that is used to allow a single system to maintain multiple TE LSAs. In all other respects, the TE LSA header is identical to the standard LSA header. The body of the TE LSA consists of a single object that conforms to the Type/Length/Value (TLV) format. The OSPF-TE draft defines two such objects, which are the Router Address TLV and the Link TLV. Each TLV contains one or more nested sub-TLVs. The Router Address TLV specifies a stable IPv4 address associated with the originating node that is reachable if the node has at least one active interface to the rest of the network. This address must not become unreachable if a set of interfaces goes down.

Nine Link TLV sub-TLVs are defined in the OSPF-TE extensions draft. They are Link Type, Link ID, Local Interface IP Address, Remote Interface IP Address, TE Metric, Maximum Bandwidth, Maximum Reservable Bandwidth, Unreserved

Bandwidth, and Administrative Group. The Link Type and Link ID sub-TLVs must appear exactly once in the Link TLV. The Link Type sub-TLV currently identifies the link as point-to-point or multiaccess, but it can support up to 255 link types. The Link ID sub-TLV carries a 32-bit field that identifies the entity to which the sourcing node is connected via the link in question; if the connection is a point-to-point link, it is the Router ID of the neighboring node.

The other sub-TLVs carried by the Link TLV are optional but none can appear more than once. Of particular interest are those related to GMPLS TE operations. The TE Metric sub-TLV carries a 32-bit TE link metric, which is not neccessarily the same as the OSPF link metric. The TE metric could be the link distance in meters, for instance, which is an important consideration when determining reach for connections over transparent optical domains. The Maximum Bandwidth sub-TLV specifies the link capacity in units of bytes/sec using the 32-bit IEEE floating point format; the Maximum Reservible Bandwidth sub-TLV likewise uses a floating-point metric in units of bytes/sec to specify the maximum bandwidth that can be reserved on the link in the direction leading from the source router; this can be larger than the Maximum Bandwidth in certain cases (such as when the link is oversubscribed). The Unreserved Bandwidth sub-TLV consists of eight floating point values that describe the free bandwidth (in units of bytes/sec) that is available at each of eight priority levels, with level 0 appearing first in the sub-TLV and level 7 appearing last.

The GMPLS extensions draft [20] defines the following four additional sub-TLVs for the Link TLV: Link Local/Remote Identifiers, Link Protection Type, Interface Switching Capability Descriptor, and Shared Risk Link Group. We describe two of these sub-TLVs here. The Link Protection Type sub-TLV is used to indicate the type of recovery scheme that is being used to protect traffic on the advertised link, or to indicate that the advertised link is serving as a backup resource for protected traffic on another TE link. Protection information is encoded in a set of flags that are contained in the first eight bits of the sub-TLV. The Shared Risk Link Group (SRLG) sub-TLV is used to identify the set of SRLGs to which the advertised link belongs. A given TE link may belong to multiple SRLGs.

B. Link Bundling

The designers of the GMPLS architecture faced a significant scaling issue in the case of core optical networks, which consist of large switches that have hundreds of fiber ports, and whose fiber links may carry hundreds of wavelengths on each fiber. A pair of switches may be connected by dozens of fibers; it is impractical to transmit a separate LSA for each one of these links. In a large network, the control plane traffic overhead would be considerable.

Since the desire is to minimize message overhead, the concept of link bundling was introduced in the MPLS working group [21]. Link bundling extends the TE link concept by addressing a problem that occurs in the creation of some types of TE links. In some cases, the elements of a TE link can be unambiguously identified using a ⟨Link_ID, label⟩ ordered pair. An example of this would be a TE link associated with a fiber port of a FSC switch whose labels correspond to the wavelengths on the fiber. In other cases, the TE construct does not produce unambiguous resource identifiers. Consider the case where the FSC switch we just described has bundled all its fiber links into a single TE link, while using the same wavelength identifiers for labels at each port. In this case the link bundling construct is required to resolve the ambiguity. Elements of bundled links are identified by an ordered triple of the form ⟨Bundle_ID, Component_ID, label⟩. The set of component links that compose a bundle are a minimal partition of the bundle that guarantees an unambiguous meaning for each ⟨Component_ID, label⟩ ordered pair. Examples of link bundling are shown in Figure 7. In Figure 7(a), two FSC nodes connected by a large number of bidirectional parallel links aggregate all of them into a single TE link that can be advertised using OSPF-TE. In Figure 7(b), we show two nodes in a BLSR/2 network connected by two unidirectional fibers, where half of the bandwidth on each fiber is available for normal traffic and the other half of the bandwidth is reserved for protection. Since both nodes are LSC the network operator chooses to aggregate all the working wavelengths on each fiber into a single TE link and to do the same for all the protection wavelengths on each fiber.

All component links in a link bundle must have the same OSPF link type, TE metric, and resource class. There are 10 link bundle traffic parameters that can be advertised by OSPF-TE. In addition to describing the link type and metric, they also describe the current bandwidth usage and the interfaces at each end of the link bundle.

C. Advertising Physical Layer Information in Routing Updates

Optical networks, because of the constraints imposed by the physical layer, are fundamentally different from the packet-switched networks for which routing protocols like OSPF were designed. The limitations on lightpath reach that result from linear and nonlinear fiber impairments, as well as coupling losses in transparent network elements, demand a more sophisticated approach to routing than the minimum hop calculations that are typically used to route IP packets. For this reason the IETF's IP/Optical (IPO) working group developed a set of input guidelines for extending link-state routing to optical networks [22]. This document, which at the time of this writing is in the RFC Editor's queue, does not specify how the relevant physical layer parameters are to be encoded in OSPF LSAs. Rather, it describes how a network operator may choose to encode information that the control plane can use to establish optical paths and lists some of the constraints that must be taken into consideration when doing so.

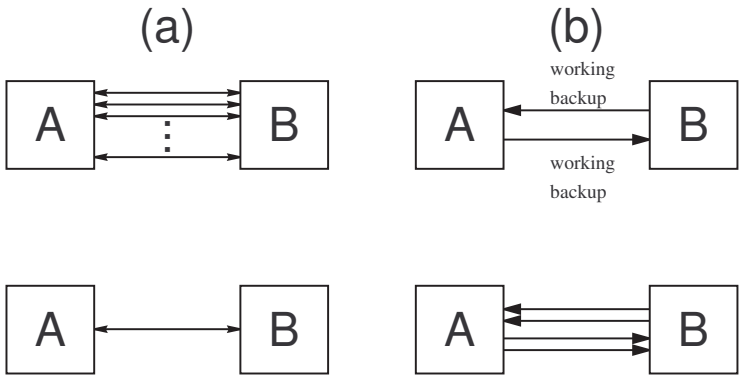


Fig. 7. Two examples of link bundling.

Physical impairments become an issue when we consider optical networks in which it is possible to create a light path where the distance between successive regeneration points is great enough to allow the optical signal quality to degrade below the minimum acceptable level. While this is less of a concern in networks that feature opaque optical switches, the transition to higher data rates and the introduction of transparent switching technologies (e.g., MEMS) means that this problem will become more acute in the future.

The draft [22] defines a domain of transparency to be an optical subnetwork in which amplification of optical signals may take place, but not regeneration. When a lightpath is created across the domain, it must be routed so that physical layer impairments are kept within acceptable limits. Transparent optical systems introduce both linear and nonlinear impairments to the signals that move through them. The draft considers how to encode linear impairments and does not recommend using LSAs to explicitly advertise information about nonlinear impairments, because they arise from complex interactions between different signals on a fiber. The best way to deal with nonlinear impairments may be to quantify the penalty that they impose and factor this into the optical link budget when computing lightpath routes.

Examples of linear impairments are polarization mode dispersion (PMD) and amplifier spontaneous emission (ASE). While PMD is not an issue in networks with widely deployed dispersion compensation devices and fibers, it can present a significant problem in legacy networks with inadequate compensation. In order for the PMD to fall within acceptable limits, the time averaged differential group delay (DGD) between two orthogonal polarizations must be less than the quantity a/B, where B is the bit rate and a is a parameter that depends on the margin associated with PMD, the outage probability, and the sensitivity of the receiver. As transmission rates increase, PMD will become more of an issue, particularly on legacy systems which may not have the compensating devices that are present in greenfield deployments. DGD is measured in units of psec/$\sqrt{\text{km}}$; the PMD on a fiber span of length L kilometers, measured in psec, is found by taking $L \cdot \text{DGD}^2$. This parameter can be encoded and distributed by the routing protocol and used by the control plane to ensure that the PMD associated with particular lightpath is below the maximum acceptable value. The other impairment, ASE, increases the probability of bit error by introducing noise into the optical signal whenever it passes through an optical amplifier. It can be encoded for each link as the sum of the noise on the link's component fiber spans. Using this information, the control plane can choose a route such that the optical signal-to-noise ratio is greater than the minimum acceptable value as defined by the operator. Because PMD and ASE do not change rapidly over time, they could be stored in a central database. However, impairment values often are not known to the degree of precision that would be required to make accurate routing decisions. Hence, improved measurement and recordkeeping capabilities are needed in order to properly use this information.

Alternatively, the network operator can use a maximum distance routing constraint, which depends on the data rate. Many carriers do this now and may choose to extend the practice to the transparent optical networks that they will deploy in the future. It can be used as the sole constraint if PMD is not an issue (either because the operator has deployed compensation devices or low-PMD fiber) and if each fiber span has the same length or the control plane handles variable-length spans by adjusting the maximum distance upper bound so that it reflects the loss associated with the worst span in the network. If this approach is used, the only information that needs to be disseminated by the routing protocol is the length of each link.

An additional issue involves routing of lightpaths to guarantee diversity, so that damage to the network's physical resources, whether caused by nature or humans (inadvertently or maliciously), has a minimal impact on the network operator's customers. To support this capability, standards development groups such as the IETF have employed the concept of the Shared Risk Link Group (SRLG), introduced in [23]. Two network elements belong to the same SRLG if they are physically co-located (e.g., two fibers that lie within a common fiber bundle would probably be assigned to the same SRLG). The use of SRLGs imposes an additional constraint when routing multiple lightpaths for a customer, namely that no pair of lightpaths should share any SRLGs.

The approach proposed in [22] is to advertise two parameters for each SRLG that capture the relationship between the links that compose the SRLG. Those two parameters are Type of Compromise and Extent of Compromise. The Type of Compromise parameter would capture the degree of closeness of the SRLG elements (e.g. shared cable, shared right of way (ROW), or shared office). The Extent of Compromise parameter would give the distance over which the closeness encoded by the Type of Compromise parameter exists.

V. Link Management Protocol

In order for the optical network to propagate signaling and routing messages, a set of control channels must be set up to link the various optical switches together and to facilitate the management of the TE data links that connect them. A key feature of GMPLS optical networks is that the control plane can be physically decoupled from the data plane by creating control channels that do not share any physical resources with the data channels that they supervise. In GMPLS, control channels can be created only when it is possible to terminate them electrically and when their endpoints are mutually reachable at the IP layer.

The Link Management Protocol (LMP), defined in [24], provides a mechanism for creating and managing multiple control channels between pairs of GMPLS nodes. It supports neighbor discovery by allowing connected nodes to verify the proper connection of their data links and to correlate those links' properties. LMP can also be used to support fault management, and thus plays a role in supporting protection and restoration capabilities in GMPLS optical networks.

A. LMP Operations

LMP operates by transmitting control messages that are encapsulated in UDP packets. Like many other protocols, LDP uses a common header that indicates the type of message being transmitted as well as the total length of the message. The message itself comprises multiple LMP objects, each of which has a standard TLV format. The first bit in each object is used as a flag to indicate whether the object is negotiable or non-negotiable.

Because LMP uses an unreliable transport layer protocol, it must incorporate a mechanism that guarantees that messages are properly received. LMP messages that initiate a process (e.g., the Config message that initiates control channel setup) carry a Message_ID object that contains a 32-bit sequence number that is incremented and wraps when it reaches its maximum value. When the recipient of a message that carries a Message_ID object sends a reply, the reply must carry a Message_ID_Ack object that contains the same 32-bit sequence number that the original Message_ID object carried.

1) Control Channel Management: In order for an LMP adjacency to exist between two nodes, there must be at least one functioning control channel established between the nodes. A control channel is bidirectional and consists of two unidirectional connections that are logically associated by the channel's endpoints. LMP creates control channels by exchanging messages that allow the nodes at the control channel's termination points to discover the IP addresses of the destination for each unidirectional control traffic flow. So that two nodes that share an LMP adjacency can distinguish between multiple control channels, each control channel destination point is assigned a unique 32-bit identifier by the node that owns it.

Four message types are used by LMP to create and maintain control channels. They are Config, ConfigAck, ConfigNack, and Hello. The first three messages are used to advertise and negotiate control channel parameters. The Hello message is used to support a fast keep-alive function that enables LMP to respond to control channel failures within the keep-alive decision cycle of the link-state routing protocol that the network is using. Hello messages do not need to be used if other mechanisms for detecting control channel failures (e.g., link layer detection) are available.

Control channel setup begins with the transmission of a Config message by one of the channel end points. In addition to identifying the control channel ID at the transmitting node, the Config message carries a set of suggested parameter values for the fast keep-alive mechanism. If the suggested values are acceptable, the receiving node will respond with a ConfigAck message that carries the 32-bit ID number for its control channel. Unacceptable parameter values are handled by transmitting a ConfigNack message with alternative parameter values that can be inserted into a new Config message to be sent by the original transmitting node as part of a new attempt. Once a ConfigAck message has been successfully received by the node that transmitted a Config message, the control channel is considered established, and both endpoints can begin transmitting Hello messages.

LMP also supports moving control channels to an inactive, or "down," state in a graceful manner. The common header for LMP messages contains a flag that can be set to indicate that the control channel is going down; a node that receives an LMP message with this flag set is expected to move the unidirectional control channel to the down state and may stop sending Hellos.

2) Link Property Correlation: The link property correlation function is used to guarantee consistency in TE link assignments between nodes that have an LMP adjacency. Link property correlation is initiated by transmitting a LinkSummary message over the control channel associated with the data links of interest. TE links can be identified using IPv4 or IPv6 addresses, or they can be unnumbered. Similar addressing options are used for data links. Data links are also identified by the interface switching type that they support, which we listed in Section II-B, and by the wavelength that they carry. LinkSummary messages are also used to aggregate multiple data links into TE links or to modify, exchange, or correlate TE link or data link parameters. If

the recipient of a Link Summary message, upon examining its database, finds that its Interface_ID mappings and link property definitions agree with the contents of the Link Summary message, it signals this by transmitting a Link Summary Ack message. If there is disagreement, a Link Summary Nack message is sent instead. If a node that sends a Link Summary message receives a Link Summary Nack message in reply, it is recommended that link verification, described the next subsection, should be carried out on those mappings that have been flagged as incorrect.

3) Link Connectivity Verification: The verification process begins when a BeginVerify message is transmitted over the control channel. The BeginVerify message establishes parameters for the verification session. The parameters include the time between successive Test messages, the number of data links that will be tested, the transport mechanism for the Test messages, the line rate of the data link that will carry the Test messages, and, in the case of data links that carry multiple wavelengths, the particular wavelength over which the Test messages will be transmitted.

The Test messages themselves are transmitted over the data link that is being tested, not any of the control channels. Test messages are transmitted repeatedly at the rate specified in the BeginVerify message until either a TestStatusSuccess message or a TestStatusFailure message is received from the destination node. The Test Status reply messages are repeated periodically until they are either acknowledged or the maximum number of retransmissions has been reached.

The IETF has also developed a draft that describes how LMP Test messages should be encoded for SONET/SDH systems [25]. In such an environment, Test messages can be sent over the Data Communications Channel (DCC) overhead bytes or sent over the control channel and correlated with a test pattern in the J0/J1/J2 Section/Path overhead bytes.

4) Fault Management: LMP can also be used to support protection and restoration operations through rapid dissemination of fault information over control channels. The actual detection of failures occurs at a lower layers; in a network of transparent optical switches, for instance, failures are detected by a loss of light (LOL) indication. LMP's fault management mechanism is decoupled from the network's fault detection mechanisms, and so will work over any opaque or transparent network.

LMP propagates fault information by using the ChannelStatus message. Each node that detects a fault transmits a ChannelStatus message to its upstream neighbor. Each upstream node responds by transmitting a ChannelStatusAck message to the sending node. Nodes that receive ChannelStatus messages correlate the failure information contained within. If the upstream node is able to isolate the failure, then signaling can be used to initiate recovery mechanisms.

B. LMP-WDM

In large-scale WDM systems it is sometimes desirable to manage the connections between an optical switch and the optical line system (OLS), which is responsible for transmitting and receiving photonic signals. The Common Control and Measurement Plane (CCAMP) working group has devised a set of extensions to allow LMP to run between the switch and the OLS [26]. In Figure 8 we show an example of LMP and LMP-WDM adjacencies.

The emphasis in the draft is on opaque OLSs (i.e., SONET/SDH and Ethernet ports); LMP-WDM can be extended to transparent switching but requirements for this are not yet clear. LMP-WDM supports the same four management functions as LMP. The draft defines a new LMP-WDM_CONFIG object that carries two flags indicating support for LMP-WDM extensions and whether the sender is an OLS. It also defines additional Data Link sub-objects for use in the Link Summary message that convey additional link information, such as the SRLGs that are associated with a particular data link, estimates of the bit error rate (BER) on a data link, and the length of the physical fiber span of the link. The draft also defines a new object to be used in the Channel Status message for fault localization.

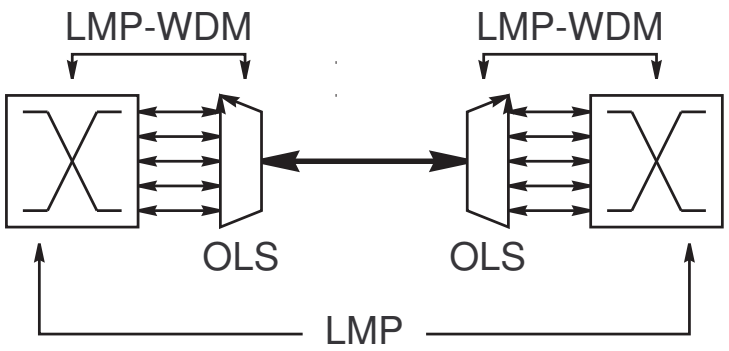


Fig. 8. LMP and LMP-WDM adjacencies.

C. LMP for Protection and Restoration

Recently, the CCAMP working group received a proposal [27] to use LMP to propagate fault information using the same type of flooding mechanism that OSPF uses to propagate link state advertisements. The proposal defines two new LMP messages,

FaultNotify and FaultNotifyAck. When a node detects a failure, it sends FaultNotify messages to each neighbor with which it shares in LMP adjacency. The FaultNotify message carries either the identifier of the failed link or a set of SRLG identifiers as well as the ID of the node that is sending the message.

The node that receives a FaultNotify message checks to see whether the failed entities identified in the message already exist in a database of failed network entities that each node would be required to maintain. If all of the failed elements are already represented in the database, the receiving node does not forward the message; otherwise it transmits a copy of the FaultNotify message to each of its neighbors. In either event, the receiving node sends a FaultNotifyAck message to the sender.

If a node that receives a FaultNotify message determines from the information that it contains that it lies on the recovery path for the traffic affected by the reported failures, it independently reconfigures its switching matrix to support the recovery path. A potential problem with this approach is the misdirection of extra traffic to unintended destinations when a switch in the middle of a protection path reconfigures itself. It is also not clear how this proposed approach will perform relative to the canonical failure reporting mechanism for GMPLS, which is targeted signaling that is confined to the working and recovery paths. These issues are being debated in the CCAMP working group at the time of this writing.

VI. Relationship to OIF and ITU-T Work

It must be kept in mind that the primary focus of the IETF with respect to optical networks is on developing extensions to IP protocols to enable IP to work well over those new types of networks. The network architecture itself is the province of the International Telecommunications Union (ITU), which has, over the past several years, generated framework recommendations that define the Automatically Switched Optical Network (ASON) [28] and Automatically Switched Transport Network (ASTN) [29]. The ITU has also developed recommendations that define the signaling network [30], routing, [31], distributed connection management [32], and automatic discovery [33].

In order to harmonize the work in these two groups, the ITU and IETF have a formal liaison relationship in which elected representatives from each group attend the others meetings and provide updates and status reports that are used, along with formal written communications, to coordinate work and request information or the undertaking of certain actions.

The other major SDO that is working in this area is the Optical Internetworking Forum (OIF). The OIF has already produced an implementation agreement that defines signaling, addressing, and other aspects of the User-Network Interface (UNI) [34]. Rather than defining new protocols, the OIF has always sought to use the machinery developed by groups such as the IETF and ITU and adapt it, where necessary, to meet the requirements enumerated by major carriers and service providers. The OIF is currently engaged in two major implementation agreement development efforts. The first, UNI 2.0, aims to define new services that will be supported across the UNI. The second effort is to develop signaling and routing for intra-carrier Network-Network Interfaces (NNIs). Both of these projects will draw extensively from the output of the IETF and the ITU.

VII. Summary

In this chapter we described the overall structure of the GMPLS control plane and the major protocol building blocks that are required to implement it. We examined the many extensions to existing IP-related protocols such as RSVP and OSPF that were required to enable traffic engineering and path management. We examined the Link Management Protocol, which was created specifically for managing control channels and verifying link connnectivity between optical network elements. We also considered how the IETF's work in this area has affected the work of other standards bodies and how various conflicts and issues are being addressed.

References

[1] Mannie, E., Ed., "Generalized Multi-Protocol Label Switching Architecture," IETF Internet Draft.

[2] Banerjee, A., Drake, J., Lang, J. P., Turner, B., Kompella, K., and Rekhter, Y., "Generalized Multiprotocol Label Switching: an Overview of Routing and Management Enhancements," *IEEE Communications Magazine*, vol. 39, no. 1, pp. 144–150, January 2001.

[3] Davie, B. and Rekhter, Y., "MPLS: Technology and Applications," Morgan Kaufmann Publishers, San Francisco, 2000.

[4] "User Network Interface (UNI) 1.0 Signaling Specification," The Optical Internetworking Forum, http://www.oiforum.com/public/documents/OIF-UNI-01.0.pdf, October 2001.

[5] "Digital Subscriber Signalling System No. 2 - User-Network Interface (UNI) layer 3 specification for basic call/connection control," ITU-T Rec. Q.2931, 1995.

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

[7] "Time To Live (TTL) Processing in Multi-Protocol Label Switching (MPLS) Networks," P. Agarwal and B. Akyol, RFC 3443, January 2003.

[8] "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Functional Description," L. Berger, Ed., RFC 3471, January 2003.

[9] "Generalized Multi-Protocol Label Switching (GMPLS) Signaling: Constraint-based Routed Label Distribution Protocol (CR-LDP) Extensions," P. Ashwood-Smith, Ed., RFC 3472, January 2003.

[10] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and Swallow, G., "RSVP-TE: Extensions to RSVP for LSP Tunnels," RFC 3209, December 2001.

[11] "Generalized Multi-Protocol Label Switching (GMPLS) Signaling: Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions," L. Berger, Ed., RFC 3473, January 2003.

[12] "The Multiprotocol Label Switching (MPLS) Working Group decision on MPLS signaling protocols," L. Andersson, G. Swallow, RFC 3468, February 2003.

[13] "Generalized MPLS Signaling–Implementation Survey," Berger, L. and Rekhter, Y., Eds., IETF Internet Draft.

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

[15] "Exclude Routes–Extension to RSVP-TE," Lee, CY, Farrel, A., and De Cnodder, S., IETF Internet Draft.
[16] Moy, J.T., "OSPF: Anatomy of an Internet Routing Protocol," Addison-Wesley, 1998.
[17] Kompella, K., and Rekhter, Y., Eds., "Routing Extensions in Support of Generalized MPLS," IETF Internet Draft.
[18] Colt, R., "The OSPF Opaque LSA Option," RFC 2370, July 1998.
[19] Katz, D., and Kompella, K., Eds., "Traffic Engineering Extensions to OSPF Version 2," IETF Internet Draft.
[20] "OSPF Extensions in Support of Generalized MPLS," K. Kompella, Ed., IETF Internet Draft.
[21] "Link Bundling in MPLS Traffic Engineering," Kompella, K., Rekhter, Y., and Berger, L., Eds., IETF Internet Draft,
[22] "Impairments and Other Constraints on Optical Layer Routing," J. Strand and A. Chiu, IETF Internet Draft.
[23] "Smart Routers–Simple Optics: An Architecture for the Optical Internet," Hjálmtýsson, G., Yates, J., Chaudhuri, S., and Greenberg, A., *IEEE/OSA Journal of Lightwave Technology*, vol. 18, no. 12, pp. 1880–1891, December 2000.
[24] "Link Management Protocol (LMP)," J. Lang, Ed., IETF Internet Draft.
[25] "SONET/SDH Encoding for Link Management Protocol (LMP) Test Messages," J. P. Lang and D. Papadimitriou, Eds., IETF Internet Draft.
[26] "Link Management Protocol (LMP) for Dense Wavelength Division Multiplexing (DWDM) Optical Line Systems," A. Fredette, Ed., IETF Internet Draft.
[27] "Extensions to LMP for Flooding-Based Fault Notification," Soumiya, T. and Rabbat, R., Eds., IETF Internet Draft.
[28] "Architecture for the Automatically Switched Optical Network (ASON)," ITU-T Rec. G.8080/Y.1304, November 2001.
[29] "Requirements For Automatic Switched Transport Networks (ASTN)," ITU-T Rec. G.807/Y.1301, July 2001.
[30] "Architecture and specification of data communication network," ITU-T Rec. G.7712/Y.1703, March 2003.
[31] "Architecture and Requirements for Routing in the Automatic Switched Optical Networks," ITU-T Rec. G.7715/Y.1706, June 2002.
[32] "Distributed Call and Connection Management (DCM)," ITU-T Rec. G.7713/Y.1704, November 2001.
[33] "Generalized automatic discovery techniques," ITU-T Rec. G.7714/Y.1705, November 2001.
[34] Jones, J., "User-Network Interface (UNI) 1.0," *Optical Networks Magazine*, vol. 4, no. 2, pp. 85–93, March/April, 2003.