nr, or n). The OCh identifiers could be defined as specified in [GMPLS-SIG] either with absolute values (e.g., channel identifiers [channel ID]), also referred to as wavelength identifiers or relative values (e.g., channel spacing), also referred to as inter-wavelength spacing. The latter is strictly confined to a per-port label space while the latter could be defined as a local or a global label space. Such an OCh label space is applicable to the OTN optical channel and the pre-OTN optical channel layer.

Applications

GMPLS extensions for G.709 must support the following applications:

- When one ODU1 (ODU2 or ODU3) non-structured signal is transported into one OTU1 (OTU2 or OTU3) payload, the upstream node requests in a non-structured ODU1 (ODU2 or ODU3) signal. In such conditions, the downstream node has to return a unique label since the ODU1 (ODU2 or ODU3) is directly mapped into the corresponding OTU1 (OTU2 or OTU3). When a single ODUk signal is requested, the downstream node has to return a single ODUk label.
- When one ODU2 signal is transported into an ODU3 payload, which is sub-divided into 16 ODU1 tributary slots, the ODU1 tributary slots (here, denoted A, B, C, and D with A < B < C < D) can be arbitrary selected. For instance, one ODU2 can be transported in ODU1 tributary slots 5, 12, 13, and 18. Therefore, when the upstream node requests in such conditions a composed ODU2 signal, the downstream node returns four labels, each of them representing a pointer to an ODU1 tributary slot.
- When a single OCh signal of 40Gbps is requested, the downstream node has to return a single wavelength label to the requestor node.
- When a composed OCh[4.2] signal is requested (i.e., a waveband or optical channel multiplex composed by four bit-rate identical Och signal of 10Gbps), the downstream node has to return four wavelength labels to the requesting upstream node since the optical channels constituting the optical multiplex are not necessarily contiguously multiplexed.

ODUk General Communication Channel (GCC)

As defined in the ITUT-G.709 recommendation, two fields of two bytes are allocated in the ODUk overhead to support two general communications channels between any two network elements with access to

the ODUk frame structure (i.e., at 3-R regeneration points). The bytes for GCC(1) are located in row 4, columns 1 and 2, and the bytes for GCC(2) are located in row 4, columns 3 and 4 of the ODUk overhead.

These bytes are defined as clear channels so that the format specification and their content can be defined for the purpose of in-fiber/in-band signaling transport mechanism.

A MORE IMMEDIATE HORIZON: GMPLS WITH SONET AND SDH

As explained in earlier parts of this book, the 3G transport network is now just beginning to emerge, and optical internets using 3G standards, and GMPLS are also in their infancy. For the more immediate future, a number of vendors and network providers think that the use of an extended GMPLS with the 2nd generation SONET and SDH transport networks can be of benefit to customers and service providers. To that end, an IETF working group has defined GMPLS extensions for use in SONET and SDH networks [MANN01]. This part of the chapter highlights the major aspects of this specification.

Traffic Parameters

The GMPLS parameters for SONET/SDH are carried in the GMPLS generalized label request within the RSVP-TE or CR-LDP packets and within the SONET/SDH payload or headers. Six fields are coded in this label request:

- Signal type (ST): This field indicates the type of elementary signal that comprises the requested LSP. Several transforms can be applied successively on the elementary signal to build the final signal being requested for the LSP. The permissible values for the signal type field are listed in Table 10–4.
- Requested contiguous concatenation (RCC): This field is used to request and negotiate the optional SONET/SDH contiguous concatenation of the elementary signal. The field allows an upstream node to indicate to a downstream node the different types of contiguous concatenation that it supports.
- Number of contiguous components (NCC): This field indicates the number of identical SONET/SDH SPEs/VCs that are requested to be concatenated, as specified in the RCC field.

Table 10–4 Signal Type Values

Value	Type
1	VT1.5 SPE / VC-11
2	VT2 SPE / VC-12
3	VT3 SPE
4	VT6 SPE / VC-2
5	STS-1 SPE / VC-3
6	STS-3c SPE / VC-4
7	STS-1/STM-0 (only when requesting transparency)
8	STS-3/STM-1 (only when requesting transparency)
9	STS-12/STM-4 (only when requesting transparency)
10	STS-48/STM-16 (only when requesting transparency)
11	STS-192/STM-64 (only when requesting transparency)
12	STS-768/STM-256 (only when requesting transparency)
13	VTG /TUG-2
14	TUG-3
15	STSG-3 /AUG-1
16	STSG-12 / AUG-4
17	STSG-48 /AUG-16
18	STSG-192/AUG-64
19	STSG-768/AUG-256
20	"VC-3 via AU-3 at the end"

- Number of virtual components (NVC): This field indicates the number of signals that are requested to be virtually concatenated.
- Multiplier (MT): This field indicates the number of identical signals that are requested for the LSP (i.e., that form the final signal). These signals can be either identical elementary signals, or identical contiguously concatenated signals, or identical virtually concatenated signals. Note that all these signals therefore belong to the same LSP.
- Transparency (T): This field indicates the type of transparency being requested. Transparency as defined from the point of view of this signaling specification is applicable only to the fields in the SONET/SDH frame overheads. In the SONET case, these are the fields in the section overhead (SOH), and the line overhead (LOH). In the SDH case, these are the fields in the regenerator section

overhead (RSOH), the multiplex section overhead (MSOH), and the pointer fields between the two. With SONET, the pointer fields are part of the LOH. Transparency is applicable only when using the following signal types: STM-0, STM-1, STM-4, STM-16, STM-64, STM-256, STS-1, STS-3, STS-12, STS-48, STS-192, and STS-768. At least one transparency type must be specified when requesting such a signal type.

SUMMARY

After reading Chapter 10, it is evident that third transport networks are intended to contain extensive MPLS operations. Other chapters have discussed the advantages of using MPLS in general, and the last two chapters have made the case for applying it in optical networks in order to exploit its TE and constrained routing capabilities. The remainder of this book brings in more information on optical networks, MPLS, and their joint contributions toward the creation of an optical Internet.

11

The Link Management Protocol (LMP)

The first part of this chapter explains the basic operations of the Link Management Protocol (LMP), including the motivation for creating the protocol. The major operations of LMP are described, as well as the messages that are exchanged between optical nodes. The second part of the chapter discusses enhancements to LMP for fault detection and recovery of optical links that run between PXCs and optical line systems (OLSs).

KEEP THE OPTICAL LINK UP AND RUNNING

Link reliability between switches and other network components has always been an important priority for the network manager. After all, if a link is inoperable, user payload cannot be sent across the link. The result is the loss of revenue and possibly the good will of the customer. The need for link reliability in an optical network is magnified by the fact that a failed link can affect many more users than if a failure occurs on a link of lesser capacity, say, a copper-wire link.

To emphasize how great the problem is, WDM systems are available in which one fiber can operate at the terabit/s rate. As noted in Chapter 1, this rate is 1,000,000,000,000, or 10^{12} bit/s. Another system multiplexes 160 wavelengths of 10 Gbit/s each for a 1.6 Tbit/s rate. Some vendors are suggesting a system that supports 320 wavelengths in a single fiber, yielding a throughput of 3.2 Tbit/s per fiber.

With these systems, a fiber failure has the potential to affect scores of millions of user connections. Therefore, there have been concerted efforts in the industry to develop link management procedures that ensure optical links stay up, and, if problems occur, to provide mechanisms for speedy recovery. One such effort, sponsored by the IETF, is called the link management protocol (LMP) [LANG01] and [BROR01]. Let's take a look at this protocol and some of its associated operations.

WHAT IS MANAGED

LMP can manage different components of the link, as shown in Figure 11–1. It can manage a data link carrying user payload, as well as control links. The term link means an optical fiber, a wavelength on the fiber, or a group of wavelengths on the fiber. LMP does not require that the control channel (or channels) be on the same physical medium as the data-bearing links; the control channel can be on a separate fiber. This common-sense approach means that the health of the data-bearing channels need to be correlated with the health of the control channel.

LMP operates on the links between optical nodes and is used for link provisioning and fault isolation. LMP is capable of handling whatever the granularity of the link may be: wavelength, waveband, or fiber. LMP operates as part of the Internet and Ethernet standards. Therefore, if it runs over Ethernet, it is identified with an Ethertype field; if it runs over PPP, it is identified with a PPP protocol ID field.

As noted, and to emphasize, a control channel is used to manage the connections between the two optical nodes. The control channel can be both in-band (part of a bundle) and out-of-band (a separate fiber).

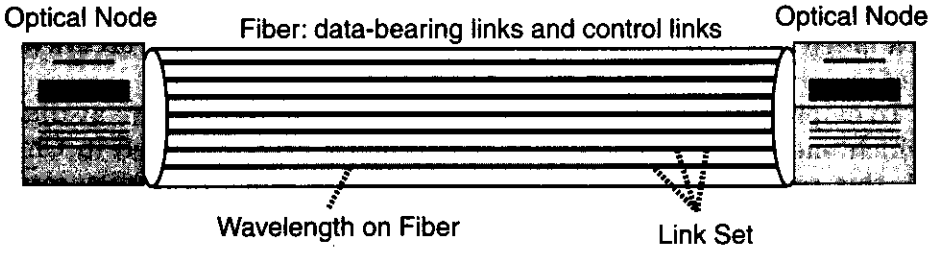


Figure 11–1 Managed components.

DATA-BEARING LINKS

An important distinction of LMP is categorizing a data-bearing link as a port or a component link. Component links are multiplex-capable, and port links are not.

LMP draws this distinction because the management of such links is different based on their multiplexing capability. For example, a SONET cross-connect with OC-192 interfaces may be able to demultiplex the OC-192 stream into four OC-48 streams. If multiple interfaces are grouped together into a single TE link using link bundling, then the link resources must be identified using three levels: TE link id, component interface id, and timeslot label. Resource allocation occurs at the lowest level (timeslots), but physical connectivity occurs at the component link level.

As another example, consider the case where a PXC transparently switches OC-192 lightpaths. If multiple interfaces are once again grouped together into a single TE link, then link bundling is not required and only two levels of identification are required: TE link id and port id. Both resource allocation and physical connectivity happen at the lowest level (i.e., port level).

LMP is designed to support the aggregation of one or more data-bearing links into a TE link (either ports into TE links, or component links into TE links).

CLARIFICATION OF TERMS

Before proceeding into a more detailed analysis of LMP, several terms used in the LMP specifications need to be clarified, and I will use LMP's terms in this chapter. The term OXC refers to all categories of optical cross-connects, irrespective of the internal switching fabric. LMP distinguishes between cross-connects that require opto-electronic conversion, called digital cross-connects (DXCs), and those that are all-optical, called photonic switches or photonic cross-connects (PXCs), also referred to as pure cross-connects. In addition, LMP refers to optical line systems (OLSs) as regenerators that are placed on long-haul links between PXCs. Even though the PXC is all-optical, long-haul OLSs typically terminate channels electrically and regenerate them optically (at least within the context of the LMP specifications), which presents an opportunity to monitor the health of a channel between PXCs [BROR01].

Because of the transparent nature of PXCs, there are new restrictions for monitoring and managing the data channels. LMP, however, can be used for any type of node.

We have learned that an all-optical node does not translate the wavelength to electrical signals, and this signal is declared by LMP to be *transparent*, and is not terminated in the node. A component link is *opaque* if it can be terminated. LMP is able to isolate faults in both opaque and transparent networks, independent of the encoding scheme used for the component links.

BASIC FUNCTIONS OF LMP

LMP is designed to provide four major functions between adjacent optical nodes: (a) control channel management, (b) link connectivity, (c) link property correlation, and (d) fault management (fault isolation). These functions are described in the following sections of this chapter.

Tables 11–1, 11–2, and 11–3 are key tools for the remainder of this chapter. The messages and fields in the messages are described in these tables, and I make frequent reference to LMP messages and fields in the following discussions.

LMP messages are "IP encoded"; that is, they use the same encoding procedure as IP. Thus, the messages can be fragmented if they are too large to meet the MTU size on the physical link by using IPv4 (or IPv6) fragmentation procedures.

LMP Messages

Table 11–1 provides a summary of the LMP messages that are exchanged between the optical nodes. The first column is the message number that resides in the message header; it identifies the message type, shown in the second column. The third column provides a short description of the function of the message.

LMP Message Header

Every LMP message contains a header, followed by the fields in the message. This header is short and simple; it contains the following fields:

- Version of LMP, which is version 1.
- Flags, which provide information on the status of the control channel, if a node is rebooting, the type of node sending the message, and if authentication fields are attached to the message.

Table 11–1 LMP Messages

Message Number	Message Type	Message Function
1	Config	Used in negotiation phase
2	ConfigAck	Acks the Config and indicates agreement on all parameters
3	ConfigNak	Indicates disagreement on Config message non-negotiable parameters
4	Hello	Sent to adjacent node to keep LMP connection alive
5	BeginVerify	Initiates the link verification process
6	BeginVerifyAck	Acks the BeginVerify message; node is ready for Test messages
7	BeginVerifyNak	Node is unwilling or unable to begin verification procedure
8	EndVerify	Terminates the link verification process
9	EndVerifyAck	Acks the EndVerify message
10	Test	Verifies the data link physical connectivity
11	TestStatusSuccess	Transmits mapping between local and remote Interface id
12	TestStatusFailure	Indicates Test message was not received
13	TestStatsAck	Acks receipt of TestStatusSuccess or TestStatusFailure
14	LinkSummary	Synchronizes Interface ids and correlates properties of the link
15	LinkSummaryAck	Acks the LinkSummary message
16	LinkSummaryNak	Naks the LinkSummary message
17	ChannelFail	Notifies a neighbor if a data link failure is detected
18	ChannelFailAck	Indicates extent of channel failure(s) in relation to received ChannelFail message
19	ChannelFailNak	Indicates that reported failures are CLEAR in upstream node (failure is isolated between two nodes)
20	ChannelActive	Indicates data link is now carrying user traffic
21	ChannelActiveAck	Acks the ChannelActive message

- The message type, as shown in column 1 of Table 11–1.
- Total length of the message, including all fields following the header.
- A checksum field to check for bit damage during the transmission.
- Local control channel id (CCid), which identifies the control channel being used for the transport of the message.

Table 11–2 Fields Residing in the LMP Messages

TLV Name	YLV Function
HelloConfig	Establishes timer values for sending Hello messages
LMP Capability	Indicates which extended LMP procedures are supported
TE Link	Indicates type of protection on the link, and its multiplexing capabilities
Data Link	Indicates if data link is a port or component link, its encoding type, and its Interface id
Failed Channel	Identifies Interface id of the failed data link(s)
Active Channel	Identifies Interface id of a data link that has become active

LMP TLVs

Like many Internet-sponsored protocols, LMP describes the formats of parts of the message with a type-length-value format (TLV). This term refers to a standard way of coding parts of the message that have variable lengths, depending on the specific message sent by an optical node to its neighbor. The TLVs follow the header, and contain: (a) type field: the specific TLV, (b) length: the length of the value field, and (c) the value field, which is the actual content of the TLV. The value field is also called the TLV object. LMP also defines one more very important field in the TLV; it is a one-bit field that indicates if a TLV object is negotiable or non-negotiable. Table 11–2 provides a summary of the TLVs currently defined in LMP.

These TLVs are explained in later parts of this chapter, but here are a few comments about some of them that should prove helpful. The LMP capability TLV is used between nodes to indicate that they are willing to execute some extra (extended) operations. The operations are: (a) the link verification procedure, (b) fault management procedure, and (c) the LMP-DWDM procedure.

The Fields in the LMP Messages

Finally, Table 11–3 provides a summary of the fields that reside in the LMP messages. Some of these fields are coded in TLVs, and others are coded as separate fields. Please note that I do not include all the fields, such as some flags that are not instrumental in understanding LMP. I refer you to [LANG01] for these details.

Table 11–3 Fields Residing in the LMP Messages

Field Name	Field Function
node id	Unique identifier for the optical node
BitRate	Rate (in bit/s) at which Test messages are sent
EncType	Optical link encoding syntaxes, such as SONET, SDH, etc.
HelloDeadInterval	How long to wait for receipt of a Hello before declaring control channel has failed
HelloInterval	How frequently Hello messages are sent
TELinkid	Id of link (local or remote)
Local Control Channel id (CCid)	Identifies control channel of the sender of the message
Message id	A unique identifier for each message, used with CCid
RcvSeqNum	Used to ack or nak Hello messages
Received Linkid	Linkid in a received message
TxSeqNum	Current sequence number in Hello message
Verify Interval	Interval at which Test messages are sent
Wavelength	Specific wavelength, measured in nanometers (nm)
LinkDown	A flag, set to 1, indicates link is down
Authentication	Several fields using MD5 for authenticating the message
Capability Flags	Indicates if link verification and fault management procedures are to be used
RemoteTELinkid	Id of TE link of remote node
VerifyTransport Mechanism	Defines transport mechanism for Test messages: (a) JO bytes, (b) DCC bytes, (c) POS, (d) GigE, (e) 10GigE
VerifyDeadInterval	If Test message not received within this value, node sends TestStatusFailure message
Verifyid	Used to differentiate Test messages from different TE links and/or LMP peers
interface id	Id of data link (port or component link)
ProtectionType	Type of link protection, such as unprotected, dedicated, shared, etc.

CONTROL CHANNEL MANAGEMENT

Control channel management is used to establish and maintain link connectivity between adjacent (neighbor) nodes. This action is accomplished using Hello messages that act as a fast keep-alive mechanism between the nodes. Before an optical link is used between two nodes, a

bi-directional primary control channel must first be configured. The control channel can be used in a variety of ways, not necessarily restricted to the manner in which LMP defines its use. For example, it can be used to exchange MPLS control-plane information such as link provisioning and fault isolation information, or path management and label distribution information using RSVP-TE, or CR-LDP. It can also be used to distribute topology and state distribution information using traffic engineering extensions to protocols such as OSPF and IS-IS.

LMP is organized along two major sets of operations: (a) control channel management is used to establish and maintain control channel connectivity between neighbor optical nodes, (b) link property correlation is used to synchronize the optical link properties between the nodes.

LMP requires that the control channel be assigned a 32-bit integer control channel identifier (CCid) to each direction of the control channel. At least one active bi-directional control channel must operate between a pair of nodes. Secondary (backup) control channels can be defined as well, and LMP provides considerable flexibility in how the secondary channels are outfitted. For example, a data-bearing channel can be preempted to become a control channel. As another example, secondary channels become active only when the primary channel is lost.

The link property correlation function aggregates multiple ports or component links into a TE link, and synchronizes the properties of the TE link between the optical nodes. As part of the link property correlation function, a LinkSummary message exchange is defined. The LinkSummary message includes the local and remote TE Link id, a list of all ports or component links that comprise the TE link, and various link properties, such as the wavelengths on the link.

All LMP messages except for the Test message are exchanged over the control channel. The Test message is sent over the data link that is being verified. Data links are tested in the transmit direction because they are uni-directional, and as such, it may be possible for both nodes to exchange the Test messages simultaneously.

Parameter Negotiation

During the control channel management operations, Config, ConfigAck, and ConfigNak messages are exchanged, and the fields in these messages are used to begin the LMP procedures. Figure 11–2 shows these procedures.

The Config message is periodically transmitted until its parameters are synchronized between the peer nodes or until one node simply

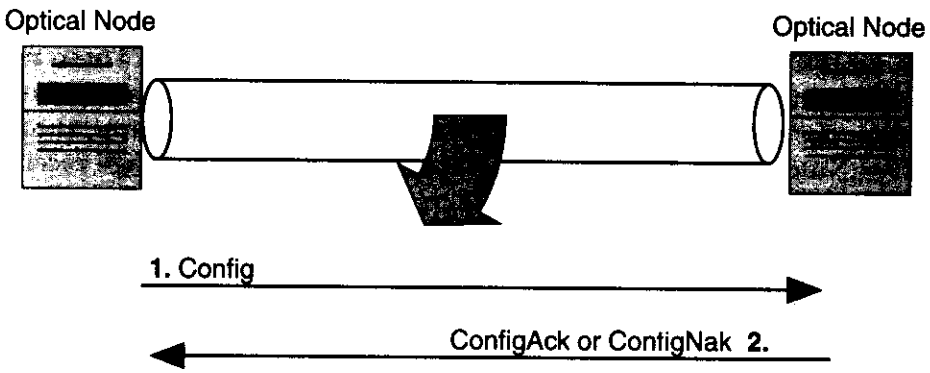


Figure 11–2 The configuration operations.

refuses to participate. The ConfigAck message is used to acknowledge receipt of the Config message and express agreement on all the configured parameters (both negotiable and non-negotiable). The ConfigNak message is used to acknowledge receipt of the Config message, to indicate which (if any) non-negotiable parameters are unacceptable, and to propose alternate values for the negotiable parameters. The Config message includes the LMP Capability TLV and the HelloConfig TLV.

The Hello Protocol

After a control channel has been established with vendor-specific configuration/crafting tools, the Hello protocol is invoked to establish and maintain connectivity between the nodes and to detect control channel failures. It is designed to react very quickly to problems so that routing protocols, such as OSPF, do not remove the adjacency status between the two nodes.

Like many Hello protocols, the LMP Hello consists of two phases: (a) the negotiation phase that establishes several operating parameters for the link, and (b) the keep-alive phase, in which the two nodes make sure that each is up and running and all is well.

Negotiation Phase. During the negotiation phase, the local and remote CCids are exchanged, and the two nodes agree on the values for the HelloInterval and HelloDeadInterval to be used on this specific control channel. These fields are encoded into the Çonfig message. The ConfigAck message acknowledges and accepts the HelloInterval and HelloDeadInterval parameters. The ConfigNak message suggests other values for these two parameters.

Fast Keep-Alive. Assuming that all these operations are agreed upon by the two nodes, Hello messages are then exchanged. These messages contain two sequence numbers. The TxSeqNum is the sequence number for this Hello message and the second sequence number (RcvSeqNum) is the sequence number of the last Hello message received from the neighbor node.

Each node increments its sequence number when it sees its current sequence number reflected in Hellos received from its peer, as shown in Figure 11–3. The sequence numbers are 32 bits in length; they start at 1

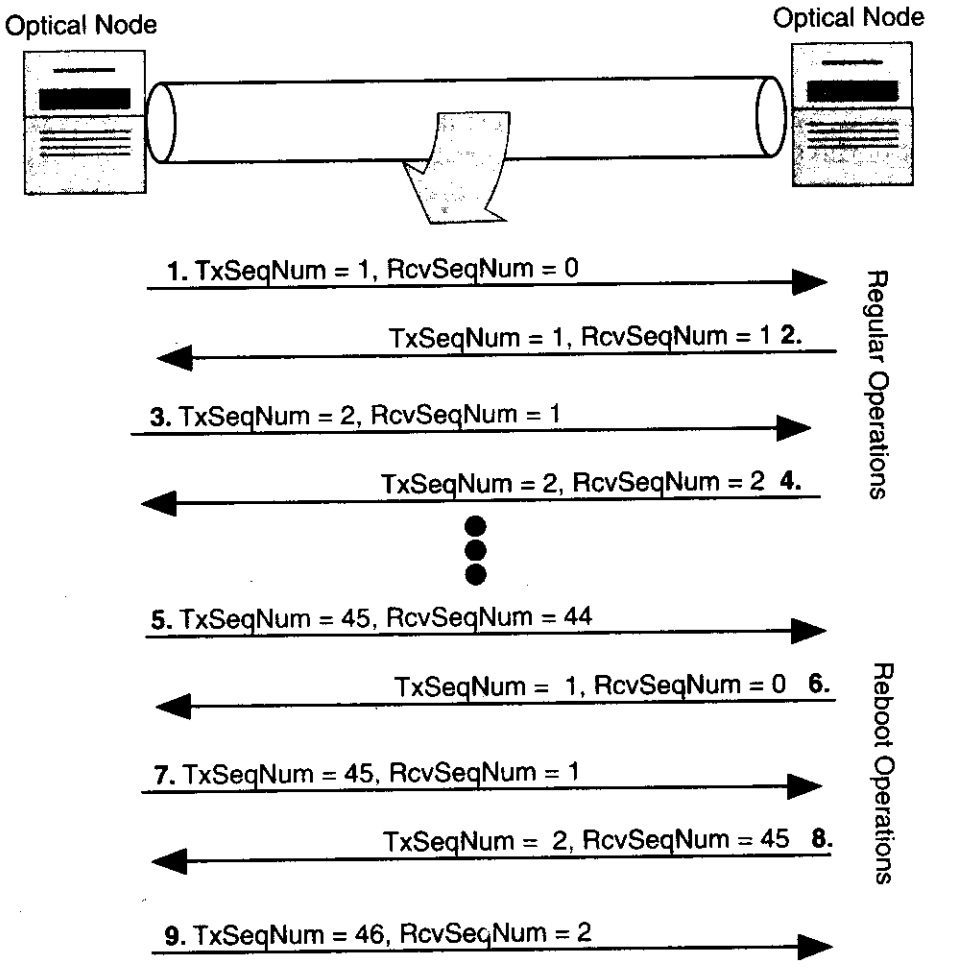


Figure 11–3 Sequencing and rebooting.

and wrap around back to 2; as shown at the bottom of Figure 11–3, 0 is used in the RcvSeqNum to indicate that a Hello has not yet been seen and 1 is used to indicate a node boot/reboot.

Under normal operations, the difference between the RcvSeqNum in a Hello message that is received and the local TxSeqNum that is generated will be, at most, 1. There are two cases where this difference can be more than 1: when a control channel reboots and when switching over to a backup control channel.

Having sequence numbers in the Hello messages allows each node to verify that its peer is receiving its Hello messages. This feature ensures that the remote node will detect that a node has rebooted if TxSeqNum=1. If this event occurs, the remote node will indicate its knowledge of the reboot by setting RcvSeqNum=1 in its Hello messages. Also, by including the RcvSeqNum in Hello packets, the local node will know which Hello packets the remote node has received, an important aspect in coordinating control-channel switchover due to a control channel failure.

Down and Degraded States. Links may be taken down for administrative or maintenance purposes. They are not taken down if component links are in use. In addition, the control channel can go down, yet the component links may be in operation. In this situation, the fiber does not have the same level of service, and it cannot accept new connections, and to bring it back up, the Config messages must be exchanged. However, if the data links are in operation, they are not taken down, but obviously they are not guaranteed the same level of service if the control channel were operable.

LINK PROPERTY CORRELATION

Link property correlation operations use the link summary messages (LinkSummary, LinkSummaryAck, and LinkSummaryNak) (a) to aggregate multiple data links (ports or component links) into a TE link, (b) to exchange interface ids, (c) to indicate the type of link protection, such as shared (M:N), dedicated (1:1), dedicated (1+1), etc. and (d) to establish the local and remote TE link id.

The LinkSummary message contains data link TLVs, and therefore might be quite large.

LINK CONNECTIVITY VERIFICATION

LMP provides an option that may be used to verify the physical connectivity of the data-bearing links (either ports or component links). Recall that in an all-optical PXC, the data-bearing links are not terminated at the PXC, but instead are passed through the switch transparently. Therefore, to ensure proper verification of data link connectivity, LMP requires that until the links are allocated, they must be opaque; that is, the bytes in them must be available for examination. To support various degrees of opaqueness of the test messages, a verify transport mechanism is included in the BeginVerify and BeginVerifyAck messages.

To interconnect two nodes, a TE link is added between them, and, at a minimum, there is at least one active control channel between the nodes. A TE link must include at least one data link (else, why bother with the exercise? . . .). The next section provides a summary of Section 5 of [LANG01].

Once a control channel has been established between the two nodes, data link connectivity can be verified by exchanging test messages over each of the data links specified in the bundled link.

To initiate the link verification process, the local node sends a BeginVerify message over the control channel. The BeginVerify message contains the number of data links that are to be verified, the interval (the VerifyInterval field) at which the test messages will be sent, the encoding scheme, the transport mechanisms that are supported, and data rate for test messages. When data links correspond to fibers, the wavelength over which the test messages will be transmitted is also included.

The BeginVerify message is periodically transmitted until (a) node A receives either a BeginVerifyAck or BeginVerifyNack message to accept or reject the verify process or (b) a timeout expires and no BeginVerifyAck or BeginVerifyNack message has been received.

If the remote node receives a BeginVerify message and it is ready to process test messages, it sends a BeginVerifyAck message back to the local node specifying the desired transport mechanism for the test messages. The remote node includes a 32-bit node unique Verifyid in the BeginVerifyAck message. The Verifyid is then used in all corresponding test messages to differentiate them from different LMP peers and/or parallel test procedures.

When the local node receives a BeginVerifyAck message from the remote node, it may begin testing the data links by transmitting periodic test messages over each data link. The test message includes the Verify

id and the local Interface id for the associated data link. The remote node then sends either a TestStatusSuccess or a TestStatusFailure message in response for each data link.

Message correlation is accomplished using message identifiers and the Verify id; this approach supports the parallel verification of data links belonging to different link bundles or LMP sessions.

When the test message is detected at a node, the received Interface id (used in GMPLS as either a Port id or Component Interface id, depending on the configuration) is recorded and mapped to the local Interface Id for that channel. The receipt of a TestStatusSuccess message indicates that the test message was detected at the remote node and the physical connectivity of the data link has been verified. The TestStatusSuccess message includes the local Interface id and the remote Interface id (received in the test message), along with the Verifyid received in the test message. When the TestStatusSuccess message is received, the local node marks the data link as UP, sends a TestStatusAck message to the remote node, and begins testing the next data link.

If the test message is not detected at the remote node within an observation period (specified by the VerifyDeadInterval field), the remote node will send a TestStatusFailure message over the control channel indicating that the verification of the physical connectivity of the data link has failed. When the local node receives a TestStatusFailure message, it will mark the data link as failed, send a TestStatusAck message to the remote node, and begin testing the next data link.

When all the data links on the list have been tested, the local node sends an EndVerify message to indicate that testing has been completed on this link. The EndVerify message is periodically transmitted until an EndVerifyAck message has been received.

Figure 11–4 shows an example of the link verification scenario that is executed when a link between PXC A and PXC B is added. The verification process is as follows:

- Event 1: PXC A sends a BeginVerify message over the control channel to PXC B, indicating that it will begin verifying the ports.
- Event 2: PXC B receives the BeginVerify message and returns the BeginVerifyAck message over the control channel to PXC A.
- Event 3: PXC A begins transmitting periodic test messages over the first port (Interface id=1).
- Event 4: PXC B receives the test messages and maps the received Interface id to its own local Interface id = 10.

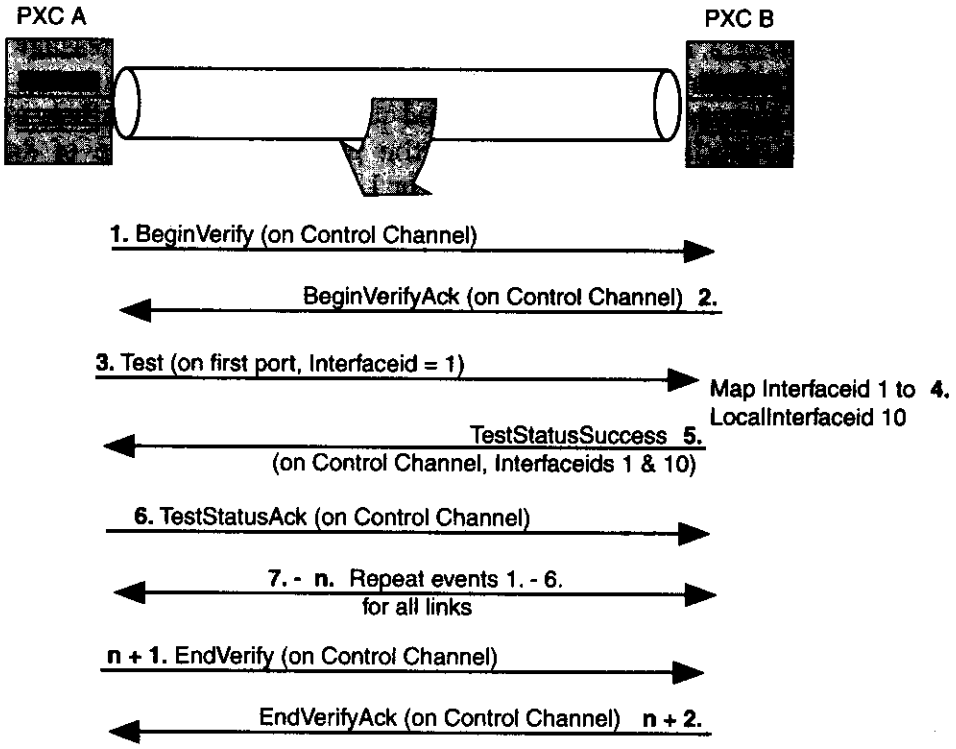


Figure 11–4 Verifying link connectivity.

- Event 5: PXC B transmits a TestStatusSuccess message over the control channel back to PXC A. The TestStatusSuccess message includes both the local and received Interface ids for the port.
- Event 6: PXC A sends a TestStatusAck message over the control channel back to PXC B, indicating that it received the TestStatus-Success message.
- Events 7 – n: The process is repeated until all of the ports are verified.
- Event n+1: Consequently, PXC A sends an EndVerify message over the control channel to PXC B to indicate that testing is complete.
- Event n + 2: PXC B responds by sending an EndVerifyAck message over the control channel back to PXC A.

FAULT MANAGEMENT

Fault isolation is used to localize failures in both opaque and transparent networks. In the situation where the nodes are O/O/O PXCs, the conventional layer 2 methods for link monitoring (framing, FCS checks, etc.) are not appropriate, and the fault detection must occur at layer 1 with the analysis of the light signals, such as the loss of light (LOL). When a failure is detected, the downstream node sends a ChannelFail message to the upstream node, which identifies all the failed component links and the associated ports.

In order to isolate the failure, the following procedures take place. An upstream node that receives the ChannelFail message will correlate the failure to see if there is a failure on the corresponding input and output ports. If there is also a failure on the input port(s) of the upstream node, the node will return a ChannelFailAck message to the downstream node (bundling together the notification of all the component links), indicating that it too has detected a failure. If, however, the fault is CLEAR in the upstream node (e.g., there is no LOL on the corresponding input channels), then the upstream node will have localized the failure and will return a ChannelFailNack message to the downstream node. Once the failure has been localized, the signaling protocols can be used to initiate span or path protection/restoration procedures.

Figure 11–5 shows three examples taken from Section 7.3 of [LANG01] of how fault isolation occurs. In scenario 1, there is a failure on a single component link between PXC2 and PXC3. Both PXC3 and PXC4 will detect the failure and each node will send a ChannelFail message to the corresponding upstream node (PXC3 will send a message to PXC2 and PXC4 will send a message to PXC3). When PXC3 receives the ChannelFail message from PXC4, it will correlate the failure and return a ChannelFailAck message back to PXC4. Upon receipt of the Channel FailAck message, PXC4 will move the associated ports into a standby state. When PXC2 receives the ChannelFail message from PXC3, it will correlate the failure, verify that it is CLEAR, localize the failure to the component link between PXC2 and PXC3, and send a ChannelFailNack message back to PXC3.

In scenario 2, three component links fail between PXC3 and PXC4. It is the job of PXC4 to correlate the failures and send a bundled ChannelFail message for the three failures to PXC3. In turn, PXC3 will correlate the failures, localize them to the channels between PXC3 and PXC4, and return a bundled ChannelFailNack message back to PXC4.

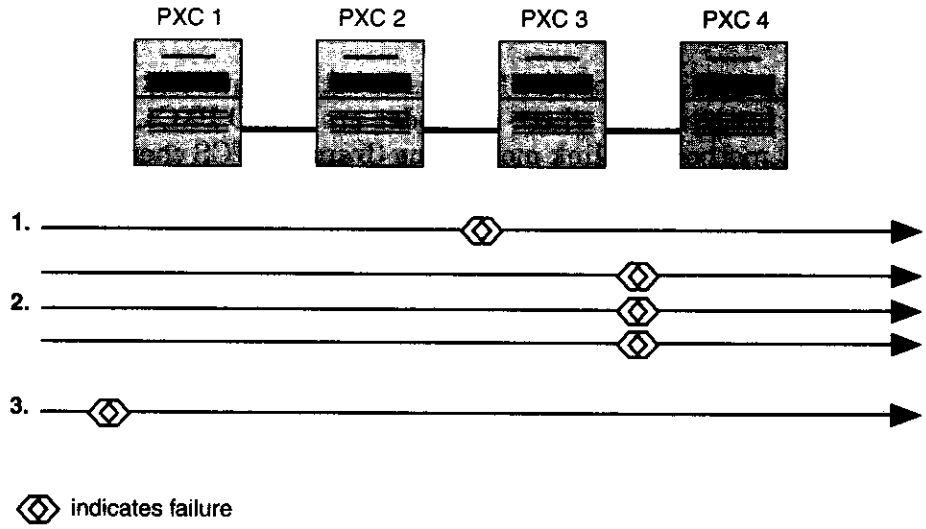


Figure 11–5 Failure scenarios.

In scenario 3, there is a failure on the tributary link of the ingress node (PXC1) to the network. Each downstream node will detect the failure on the corresponding input and send a ChannelFail message to the upstream neighboring node. When PXC2 receives the message from PXC3, it will correlate the ChannelFail message and return a ChannelFailAck message to PXC3 (PXC3 and PXC4 will also act accordingly). Since PXC1 is the ingress node to the optical network, it will correlate the failure and localize the failure to the component link between itself and the network element outside the optical network.

EXTENDING LMP OPERATIONS FOR OPTICAL LINK SYSTEMS (OLSS)

The focus of LMP is to manage the optical links between the optical switches, such as PXCs. The Internet standards groups have also recognized the need to manage links between the optical link systems (OLSs) that are installed between the switches and these OLSs. The revised editions of LMP do provide sufficient rules to implement link management at OLSs. This section of the chapter explains an extension to LMP to support OLSs and highlights the salient parts of [BROR01]. The situation is illustrated in Figure 11–6. As discussed thus far in this chapter, LMP

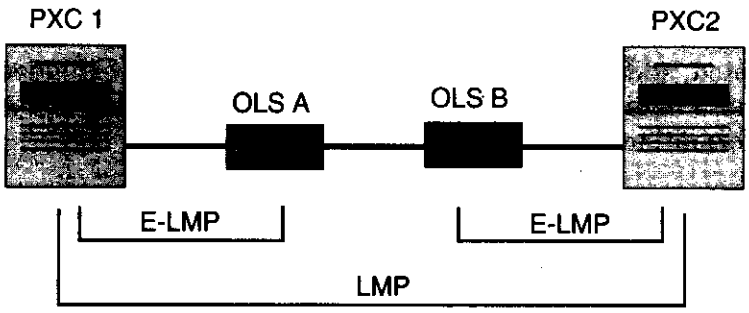


Figure 11–6 Extended LMP (E-LMP).

has been designed to operate between PXCs, or other types of optical switches. Extended LMP operates between the switch and the OLS.

The idea for extending LMP to the OLS is to use the information the OLS has on the activity on the links between the switches. Providing this information to the control plane via LMP can improve network usability by further reducing required manual configuration and also greatly enhancing fault detection and recovery. Even though the PXC is all-optical, long-haul OLSs may terminate channels electrically and regenerate them optically (although this situation will change), which presents an opportunity to monitor the health of a channel between PXCs.

However, extended LMP is not confined to the PXC/OLS operation. It can be applied to any optical link that exhibits opaqueness. I think the authors should retitle their IETF draft, because the limited title does not do justification to the specification.

Because extended LMP is based on LMP, the best approach is to refer to the major LMP functions discussed earlier in this chapter and look at how extended LMP uses them. You may want to refer back to the LMP material during this discussion.

- Control channel management: Extended LMP uses the same rules as LMP.
- Link verification: Extended LMP uses the same test procedures as LMP.
- Link summarization: Extended LMP defines additional TLVs and fields (this is explained later).
- Fault management: Extended LMP is the same as LMP, except that additional messages and procedures are defined (and explained later).
- Trace monitoring: This feature is new and is not defined in LMP.

Link Summarization

As noted, additional TLVs and fields are defined in extended LMP to provide more details about link characteristics. Following is a summary of this information:

- Link group id. Identifies a group of data links. It allows the sending of one message (instead of perhaps many) for each member of a group. For example, a link group can be created for each laser on a node or for each fiber. A failure to the laser and/or the fiber can affect many data links, and they can be identified with the link group id.
- Link descriptor. Specifies the minimum and maximum reservable bandwidth for a data link, and the encoding type for the link (SDH, SONET, etc.).
- Shared risk link group identifier (SRLG). This id is used to define the link's membership in a group of data links.
- Bit error rate (BER) estimate. Used to gauge the quality of the data link, such as 10^{12}, 10^{13}, and so on.
- Optical protection: Specifies how (if) the OLS protects the data link, with the schemes listed in Table 11–4.
- Span length: Explains the distance of the OLS fiber, expressed in meters.
- Administrative group (color): Specifies the administrative group (or color) to which the data link belongs.

Table 11–4 OLS Protection Schemes [ASHW01]

Enhanced:	A protection scheme that is more reliable than Dedicated 1+1 should be used, e.g., 4-fiber.
Dedicated 1+1:	A dedicated link layer protection scheme, i.e., 1+1 protection, should be used to support the LSP.
Dedicated 1:1:	A dedicated link layer protection scheme, i.e., 1:1 protection, should be used to support the LSP.
Shared:	Indicates that a shared link layer protection scheme, such as 1:N protection, should be used to support the LSP.
Unprotected:	The LSP should not use any link layer protection.
Extra Traffic:	The LSP should use links that are protecting other (primary) traffic. Such LSPs may be preempted when the links carrying the (primary) traffic being protected fail.

Fault Management

The overall fault management operations between LMP and extended LMP are similar. The main differences are that the OLS can initiate upstream and downstream testing, and it does not participate in end-to-end fault localization that LMP performs. Since OLS has more detailed information about an optical link, it can more concisely identify faults, say, of amplifiers on the optical span.

Extended LMP uses several ChannelStatus messages (requests, responses, Acks, Naks) on the control channel to report the status of the data link. It must be sent each time the status of a channel changes. The message contains a condition field that is coded to reveal the following condition of the data link:

- Signal OK: Data link is operational.
- Signal degrade: BER has exceeded a threshold, typically in the range of 10^{-5} to 10^{-9}.
- Signal fail: Indication of a SONET/SDH-defined hard failure, such as LOS, LOF, Line AIS, or a BIP exceeding a threshold.

Trace Monitoring

This extended LMP capability is used to request an OLS to monitor one or more data links for a specific event. It uses several TraceMonitor messages (requests responses, Acks, Naks) to define the type of trace and, of course, to respond to the request for the trace type. The trace types defined thus far are:

SONET section trace (J0 byte)
SONET path trace (J1 byte)
SDH section trace (J0 byte)
SDH path trace (J1 byte)

SUMMARY

Scores of link management protocols have been invented during the past few decades. All are concerned with managing the communications links between nodes such as servers, routers, and host computers. The LMP is yet another example of a link management protocol, and it is tailored to operate on and support optical links. It is capable of managing the link itself, as well as the wavelengths on the link.

12

Optical Routers: Switching in Optical Internets

This chapter discusses how optical routers forward IP packets through the optical network. The term router is used in more than one context. It can be a conventional IP-based node, a label switching router (LSR), an O/E/O node, an O/O/O node, or a combination of any of these attributes. The specific role the router plays depends on its position and responsibility in the network, and this role will be identified as appropriate.

We also look at the relationships between a label switching path (LSP) and an optical switched path (OSP), the architecture of a Micro-Electro Mechanical System (MEMS), and the role of MPLS and optical cross-connect tables in switching and protection switching operations.

THE STATE OF THE ART IN OPTICAL SWITCHING

Parts of this chapter look to the future. First, optical switches are far from mature, and the demand for them at this time is limited. Second, 3rd generation transport internets using PXCs are not yet implemented. Third, work is not complete on specifications for IP, MPLS, and lambda control plane interworkings, not to mention that there are no implementations other than in labs. Nonetheless, this book is part of the series titled Advanced Communications Technologies, so, to that end, let's examine how the optical Internet can evolve with regard to switching operations.

ORDER OF PREFERENCES IN SWITCHING IMPLEMENTATIONS

Throughout this chapter, it will be a goal to resort to layer 1 switching, called lambda switching. Pure lambda switching is not yet possible unless the switched wavelength remains the same through the switch. As of the writing of this book, the prevalent photonic switch in the industry is opaque and must use transponders for O/E/O operations. However, we will see that this restriction need not preclude the use of O/O/O data plane operations for certain paths and traffic.

As a general guideline, the preferences (from least desirable to most desirable) are listed below, and the evolving networks will use combinations of these methods:

1. O/E/O with IP forwarding
2. O/E/O with label switching
3. O/E/O with λ switching (and wavelength conversion)
4. O/O/O with λ switching (and no wavelength conversion)

CLARIFICATION OF KEY TERMS

To make certain two key terms are understood, this part of the chapter re-emphasizes these terms:[1]

- Label switched path (LSP): The end-to-end path between two MPLS users, including all nodes and links that are on the path between these two users. An LSP segment between two adjacent nodes represents one physical path of the logical LSP. LSP segments are concatenated together to form the end-to-end LSP.
- Optical switched path (OSP): The node-to-node paths between the users of the optical link. The OSP is not an end-to-end path between the users; it exists only between two adjacent MPLS/optical nodes. The OSP begins at one node and terminates at a neighbor node that is directly adjacent to the originating node. It is the job of an MPLS-optical network to be able to concatenate each OSP segment to that of an MPLS segment, and, ultimately, to an

[1]A small point of calrification about these terms: Some literature uses the term switching instead of switched. Both terms are acceptable.

end-to-end LSP. In this chapter, we will see how this operation is accomplished.

- OSP cross-connect table: The literature on optical networks uses the terms wavelength forwarding information base (WFIB), switching fabric, and optical routing table to describe this component in an optical switch. I prefer OSP cross-connect table, since this term conveys the idea of correlating an input OSP to an output OSP.

ONE AFTERMATH OF SEPTEMBER 11: INCREASING LOAD ON THE TRANSPORT NETWORKS

It was noted in Chapter 1 that the transport backbone has more bandwidth than is currently needed. Nonetheless, most predictions state that Internet traffic alone will as least double by 2003. Other predictions are even higher.

In my view, there is no question that the tragic events of September 11, the Anthrax threats, and other likely future happenings will push even more mail and correspondence onto the Internet.

Our nation's physical infrastructures are porous and fragile; such are the underpinnings of an open society. But the telecommunications infrastructure is far more immune to electronic viruses than the other infrastructures' almost helpless defenses against biological and chemical attacks. It follows that organizations and individuals will move more to electronic mail and electronic commerce. This sudden event has not yet been absorbed by the industry (I am writing this passage shortly after the September 11 tragedy).

The point of my discussion here today is that it is even more important that the transport networks have the capacity and robustness to support what I believe will be a huge increase in electronic mail, faxes, legal documents, etc.

EVOLUTION OF SWITCHING TECHNOLOGIES

One of the areas of interest in high-speed networks is how fast these networks can relay traffic from node to node. Figure 12–1 provides a summary of the evolution of switching technologies. Since the inception of

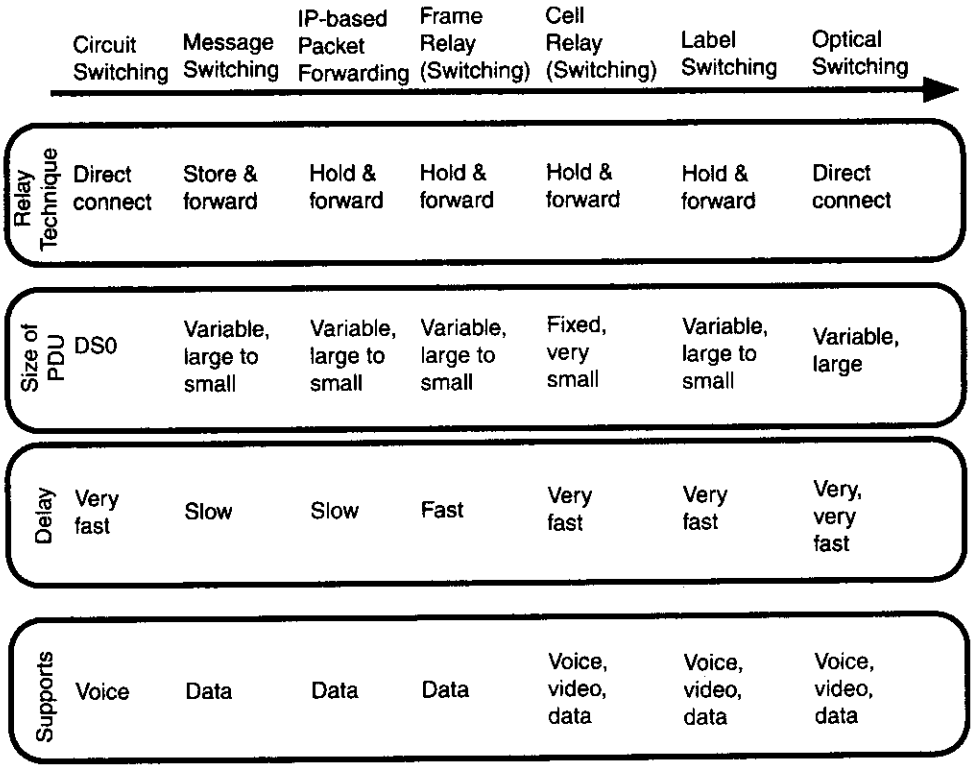


Figure 12–1 Evolution of switching/routing systems.

circuit switching in telephone networks, there has been a migration away from some of the circuit-switching concepts, principally from the reliance on the fixed TDM DS0 channel. This migration led to message switching in the 1960s, packet switching in the 1970s, and frame and cell switching in the 1980s and 1990s. Most of these technologies (except cell relay) allow the protocol data unit (PDU)[2] to be of variable length. In addition, IP packet switching introduced the idea of a connectionless network, one in which there were no circuits set up (physical or virtual), and the packets were completely self-contained with a source and destination

[2]The term protocol data unit, or PDU, was invented by the ITU-T to describe any "piece" of traffic that is transmitted across a communications link. It can refer to any layer's traffic, thus it is a convenient (and generic) term.

address in each packet. This approach allowed networks to use adaptive, dynamic relaying techniques, with the result that traffic from one user session might take different paths through the network, and arrive out of order.

The attraction of circuit switching (its low delay) has come back into vogue, but its fixed-size unit has not. In addition, it is recognized that the use of IP destination-based forwarding, as well as dynamic routing (route discovery) is not a preferred approach for modern networks. Therefore, we see an evolution toward a synthesis of some of these forwarding and relaying technologies:

- Keeping the delay through the network as low as possible.
- Keeping jitter (variable and accumulated delay) consistent.
- Allowing the use of variable data units.
- Relying on MPLS label switching to replace IP destination-based routing.

But as suggested by the right-most entry in Figure 12–1, label switching is not the end of the story. The current label switching routers (LSRs) use electronics to make their switching decisions. While these machines are very fast, they suffer from the fact that they use the O/E/O processing introduced in Chapter 1. They must convert the optical signal to an electrical counterpart, make the switching decision with electronics (and software) and then convert the outgoing signal (and packet) back to an optical signal for transferral across the next optical fiber link.

THE SPEEDS OF ELECTRONICS AND PHOTONICS

The basic problem is that the performance (in speed) of photonics is outpacing electronics—the so-called electronic bottleneck. The optoelectronic conversions create too many delay points in the network. The solution is obvious. The O/E/O switch must be redesigned to be an O/O/O switch, also called a photonic switch, or a photonic cross-connect (PXC). A vast effort is underway in the industry to invent this technology, and another part of this chapter describes these emerging switches. But for now (and the current marketplace), let's take a look at the optical router, a machine that performs O/E/O operations.

AN OPTICAL ROUTER

Figure 12–2 shows a functional diagram of a router that is capable of processing optical packets [BLUM01]. The major functions of the router are:

- Demux and mux: Separates and combines wavelengths.
- Optical splitter: Sends copies of packet to control element and/or label eraser.
- Label eraser: Removes label header.
- Label writer: Places a new label on the packet.
- Wavelength converter: Places packet onto one of several wavelengths.
- Buffer: Holds packet until mux is ready to process it (work is underway to develop techniques to support this difficult requirement).
- Control element: Controls the operations of the label writer, the wavelength converter, and the buffer.

The Control Element

It is very difficult to build an all-optical logic circuit in which light controls light. Lab experiments have demonstrated that it is possible, but for the immediate future (the early part of this decade), the optical router will have some of its control elements made of electronics.

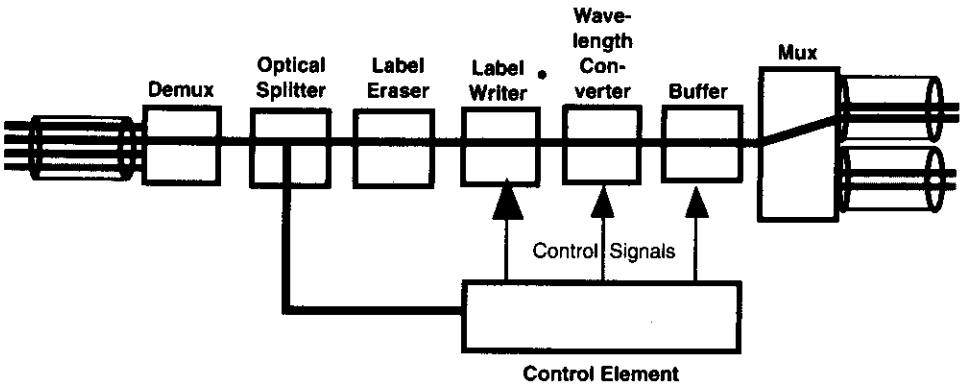


Figure 12–2 Functional view of the optical router [BLUM01].

Figure 12–3 shows two implementations of the control element in the router. The first relies on electronics. The incoming packet has some of its energy diverted to the control element. The photodetector converts the optical signal to an electronic signal and is processed by the control electronics. The incoming label in the packet header is matched to a routing table and an outgoing label is chosen. The control element instructs the switch as to which wavelength is to carry this packet. The payload is not processed by the control element; thus the optical-to-electronic conversion is performed on a very small set of bits (probably 20 bits, the size of an MPLS label).

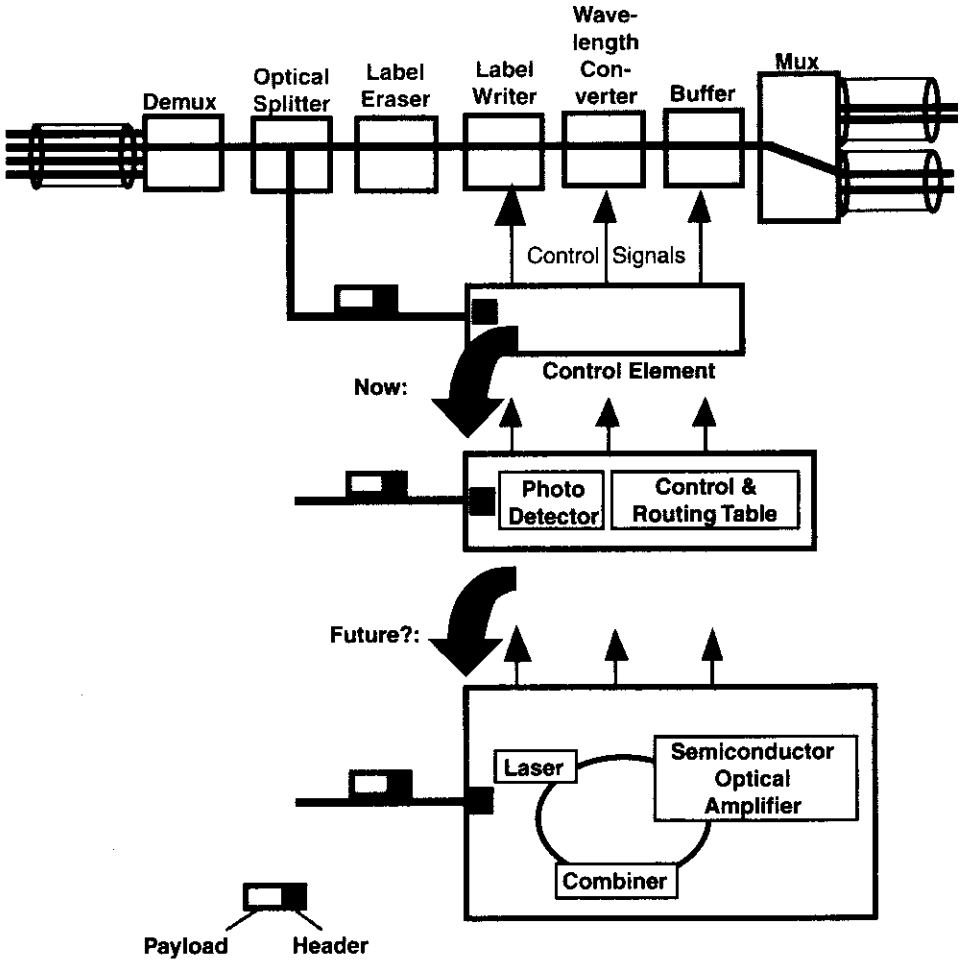


Figure 12–3 The control element [BLUM01].

The challenge (and problem) is the slow speed of electronics in relation to the amount of optical traffic that must be processed by the switch. It is recognized that the transfer of the packet from one fiber to another must eventually occur in less than one nanosecond. Several prototypes have been developed to deal with this problem. One such device is a semiconductor-amplifier, shown at the bottom of Figure 12–3. It forms a optical bridge between the appropriate input and output fibers. When the packet reaches the bridge, a control signal (an electric current) injects electrons and holes (absence of electrons) into the amplifier. When light enters the amplifier, it causes the holes and electrons to combine, giving off photons. These photons are the exact copy of the optical packet that is going to pass through the bridge. After reaching a certain power threshold, the signal moves from one side of the bridge to the other. Of course, the control of this system is with electrical signals, but the photonic packets need not be converted to electronic packets.

We continue the examination of the optical router by examining each of its other components in more detail. Figure 12–4 shows a packet arriving at the router on λ2. The packet is composed of the header, which contains the label, and the payload, which contains traffic (user traffic or control traffic).

The demux (demultiplexer) separates the wavelengths into different pathways, and the optical splitter sends the packet to the control element and the label eraser.

The demux and the optical splitter do not process the packet's contents. They simply send the packet, in its entirety, to the next components. Also, the packet at this point is still associated with a specific optical channel (in this example, the λ2 channel).

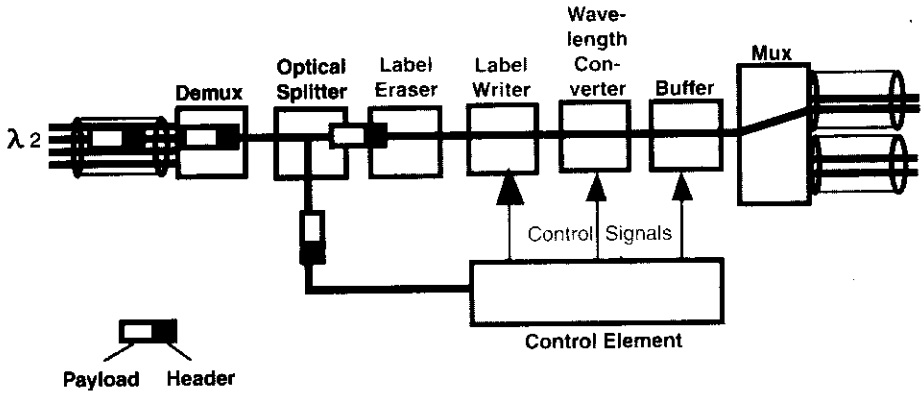


Figure 12–4 Operations at the demux and the optical splitter [BLUM01].

It was noted earlier that the control element controls the label writing operations. Figure 12–5 shows the operations of the label eraser and the label writer in conjunction with the control element. The label eraser (without interacting with the control element) removes the header from the packet, and the label writer inserts a new header. The header contains either an IP address or a label. (It would be very inefficient to use an address.)

The control element thus contains the label swapping information. This information has been loaded into the control element based on the execution of:

- IP address and discovery operations using OSPF, IS-IS, or BGP.
- Binding the discovered address to a label at this router, both for the incoming label and the outgoing label (the label that was inserted in Figure 12–5).

Again, under the direction of the control element, the wavelength converter sends the packet from one wavelength ($\lambda 2$) to another ($\lambda 3$), as depicted in Figure 12–6. Although not shown in this figure, the same operations are occurring with the other three input wavelengths.

Conceptually, the optical buffer holds a packet until the control element instructs it to send the packet to the output multiplexer. This

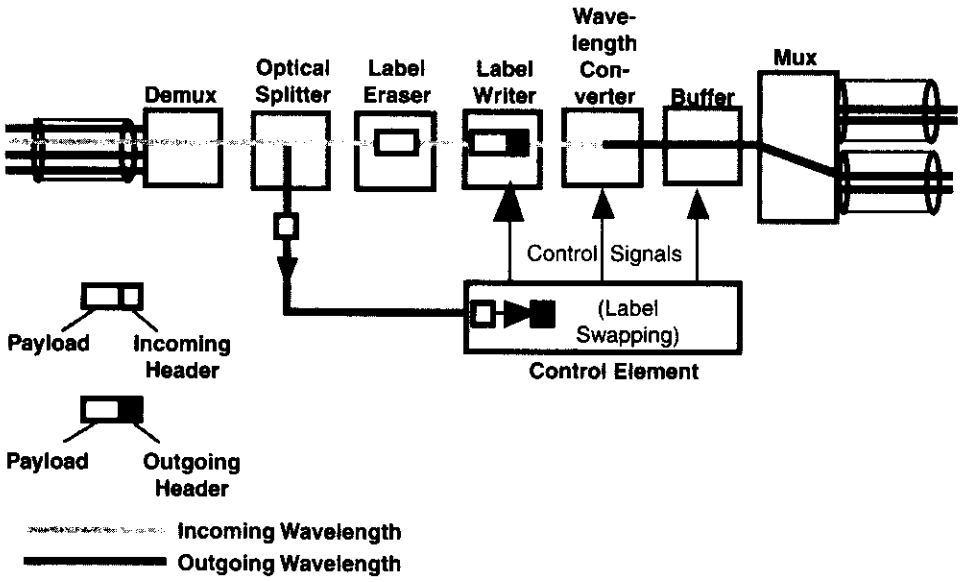


Figure 12–5 Operations of the label eraser and the label writer [BLUM01].

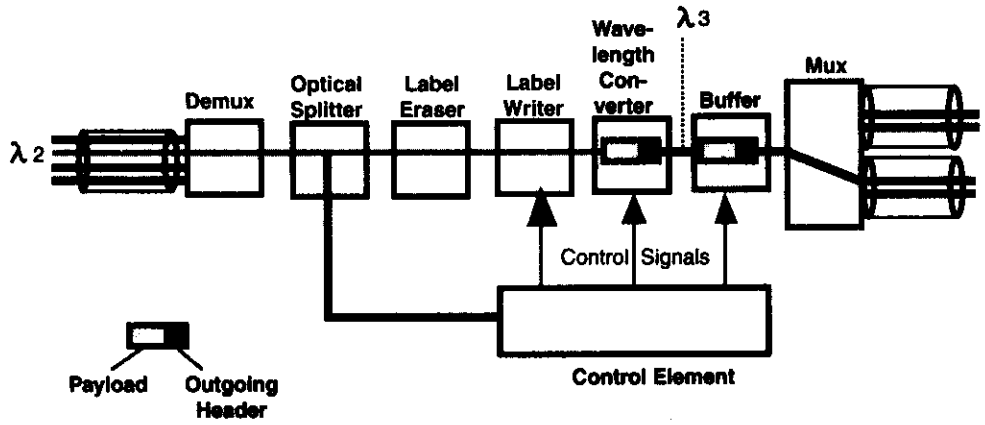


Figure 12–6 Operations at the wavelength converter and buffer.

operation is more easily described than implemented. An electronic buffer uses dynamic random-access memory (DRAM) to hold the packets until they are sent out onto the link. Photons cannot be stored like electrons. Therefore, an optical buffer may (a) "corral" the light pulses into a holding area (something like an automobile traffic circle), or (b) synchronize the time it takes to move the pulses through the switch to that of the processing time at the switch so that both end their respective movements and tasks at the same time.

The last operation at the optical router occurs at the output mux, shown in Figure 12–7. This element sends the packet on to the appropriate wavelength and the appropriate fiber.

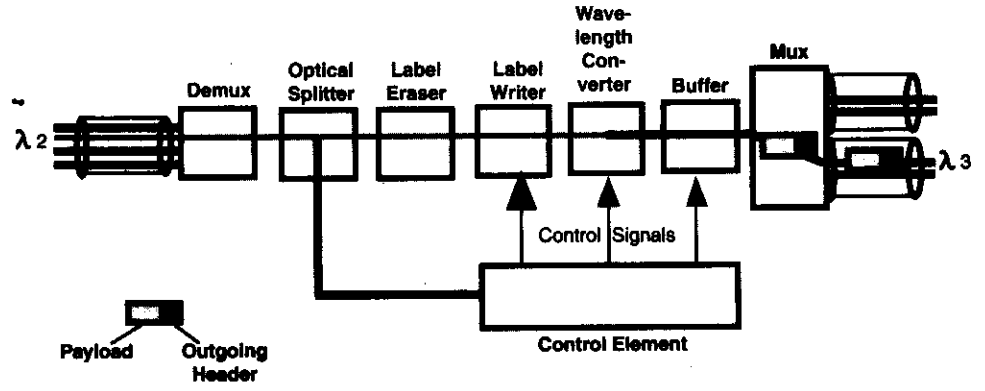


Figure 12–7 Operations at the output multiplexer.

OPTICAL SWITCHING TECHNOLOGIES

As noted earlier in this chapter, the volume of traffic in networks today, and the real-time requirements of this traffic, are creating a demand for an O/O/O (PXC) switch. The basic concept is simple: User traffic is modulated and multiplexed on to a wavelength, perhaps at the edge router of the network to an optical backbone network. Thereafter, the wavelength is transferred through the switch, dropped off, or imprinted onto a different wavelength, perhaps onto a different output fiber. At the terminating edge router (leaving the network), the wavelength is converted to electronics, and the user's traffic is taken through the conventional routing processes to deliver it to the final recipient. The next part of the chapter provides an overview of an emerging technique for optical switching [BISH01].

OPTICAL RESOURCES

For the remainder of this chapter, it is important to keep in mind that the deployment of optical networks that provision and perhaps switch high-capacity bandwidths (say OC-192) must be managed quite carefully. A single OC-192 link across a wide area is very expensive, and its efficient use and resilience is quite important. The nailing-up, switching, and tearing-down of these links can certainly be accomplished with a well-designed hardware and software platform (as will be demonstrated in the remainder of this book), but this platform must be applied judiciously.

As of this writing, it is too soon to know, but it seems likely that long-haul optical networks that integrate IP and MPLS will be designed to aggregate very large communities of users at major POPs (playing the role of ingress nodes to the long-haul backbone) into a small number of labels and wavelengths for transport across the backbone across wavelengths that tend to stay nailed up for a long time.

With these thoughts in mind, the next section examines optical switches.

MicroElectroMechanical Systems (MEMS)

MEMS is a technique used in the fabrication of the optical switch. It involves building a very small set of mirrors that are positioned to be illuminated by one or more wavelengths. In effect, the incoming wavelength is reflected by the mirror to an outgoing wavelength, as shown in Figure 12–8.

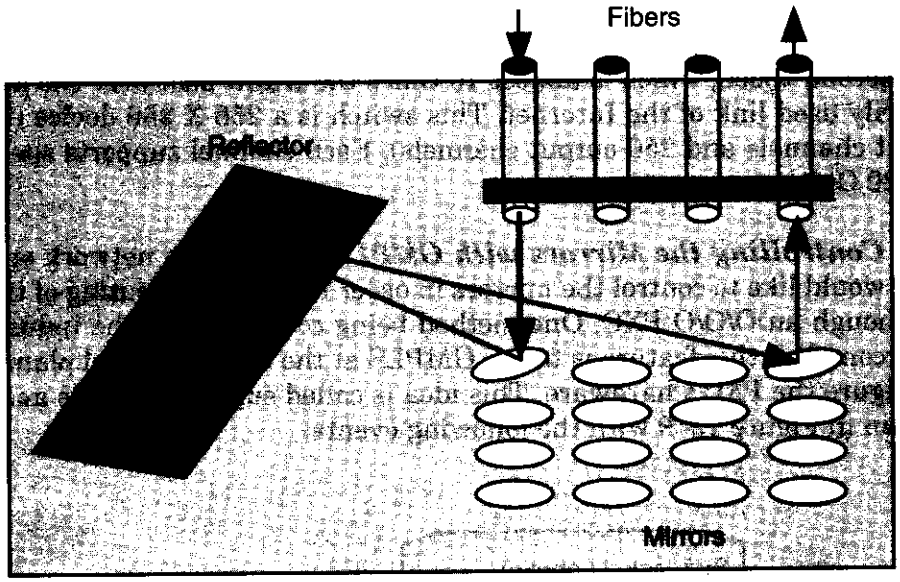


Figure 12–8 MEMS [BISH01].

The MEMS is constructed in a manner similar to the making of integrated circuits by using photolithographic and etching operations. However, in place of transistors, the lithographic process builds very small devices that move (tilt) when directed by an electric current. In effect, these devices are tiny mirrors about .5 millimeter in diameter (about the size of a pin head). The mirror is illuminated by the optic wavelengths coming from a fiber. Based on how the mirror is tilted, the wavelength can be reflected from N input fibers to M output fibers. The tilting of the mirrors is controlled by software.

One manufacturer of a MEMS switch has 256 mirrors deposited on a 2.5 centimeter-square piece of silicon. The mirrors are one millimeter apart, which translates into a switch that is 32 times more dense than an electronic switch. The switch performs a complete switching operation in a few milliseconds. Moreover, since the optoelectronic conversion is not needed, the MEMS switch provides up to a 100-fold reduction in power needs.

In an extraordinary example of the versatility and power of silicon-based technology, the MEMS technology takes advantage of how amino acids arrange themselves into three-dimensional shapes. During the final stages of the fabrication process, small springs on the surface of the silicon lift each mirror above the silicon surface to allow them to move (tilt), all under the control of software.

[BISH01] describes Lucent's LambdaRouter, the first large-scale MEMS switch announced in the industry. It operates at 10 terabits-per-second speed, which is almost 10 times the traffic sent over the most heavily used link of the Internet. This switch is a 256 X 256 device (256 input channels and 256 output channels). Each channel supports speeds of 320 Gbit/s.

Controlling the Mirrors with GMPLS. Ideally, a network operator would like to control the mirrors in order to affect the routing of traffic though an O/O/O PXC. One method being considered in the industry to accomplish this feature is to use GMPLS at the optical control plane to configure the PXC's hardware. This idea is called suggested labels and is shown in Figure 12–9 with the following events:

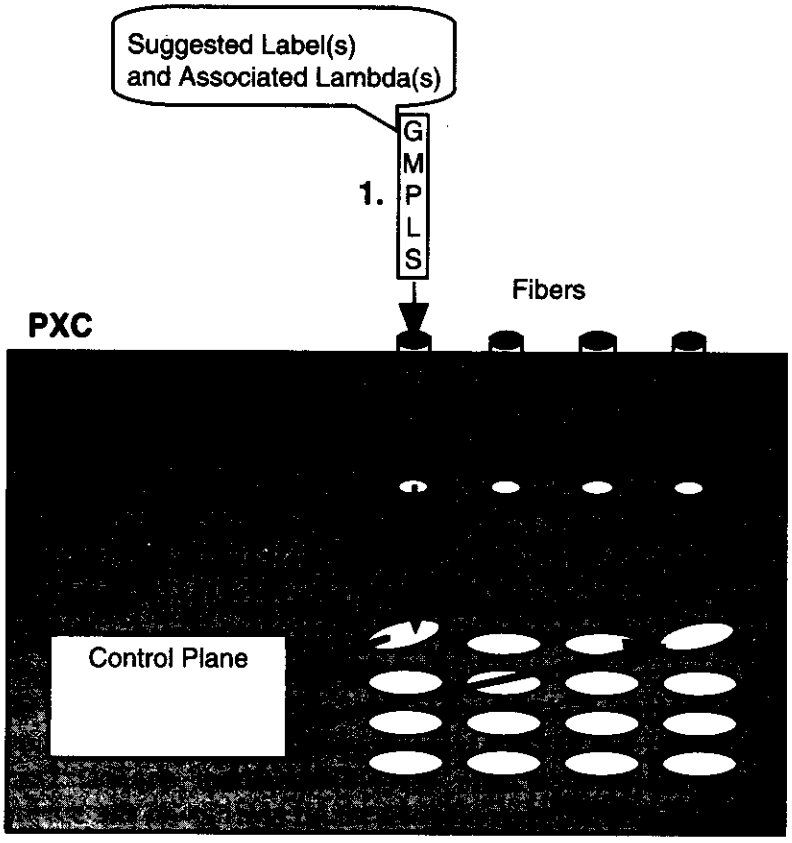


Figure 12–9 Configuring the MEMS mirrors.

- Event 1: The GMPLS message contains a suggested label for one or more wavelengths.
- Event 2: This message arrives over a designated control channel, as established with the Link Management Protocol (Chapter 11).
- Event 3: The message is sent to the control plane where software acts on the GMPLS information (such as assigning a label to a wavelength).
- Event 4: The PXC is configured with the control plane altering a mirror to change its reflecting angle, and thus its output characteristics, say, to another fiber.

You might be wondering why the term "suggested label" is used. As a general rule, MPLS stipulates that the downstream node is to specify the label. In this example, the PXC is indeed a downstream node, but it is accepting a label suggested by the upstream node (not shown in the figure). GMPLS recognizes that the suggested label by the upstream PXC allows the upstream PXC to begin its configuration as soon as it sends the suggested label to the downstream node. In this manner, the upstream node can begin early configuration of its hardware to reduce configuration delays. However, in keeping with the basic MPLS approach, if the downstream PXC passes a different label to this peer upstream node, the upstream node must use it.

If you would like to read more about MEMS, both its potential and its problems, I recommend [FERN00].

PROTECTING THE LABEL SWITCHED PATH

In the event that a link or node fails, a robust network must provide protection to the customer's traffic that is traversing the part of the network that encounters problems. We introduced this idea in Chapter 8. In this section, we extend the information in Chapter 8 with examples of how a partnership between MPLS and an optical switch can help protect the user's payloads. The analysis begins with a typical MPLS protection scenario.

MPLS supports the concept of protection switching and backup routes. An MPLS network can be set up to assure that a link for node failure will not create a situation where the user traffic is not delivered. Figure 12–10 shows the operations to recover from a failure.

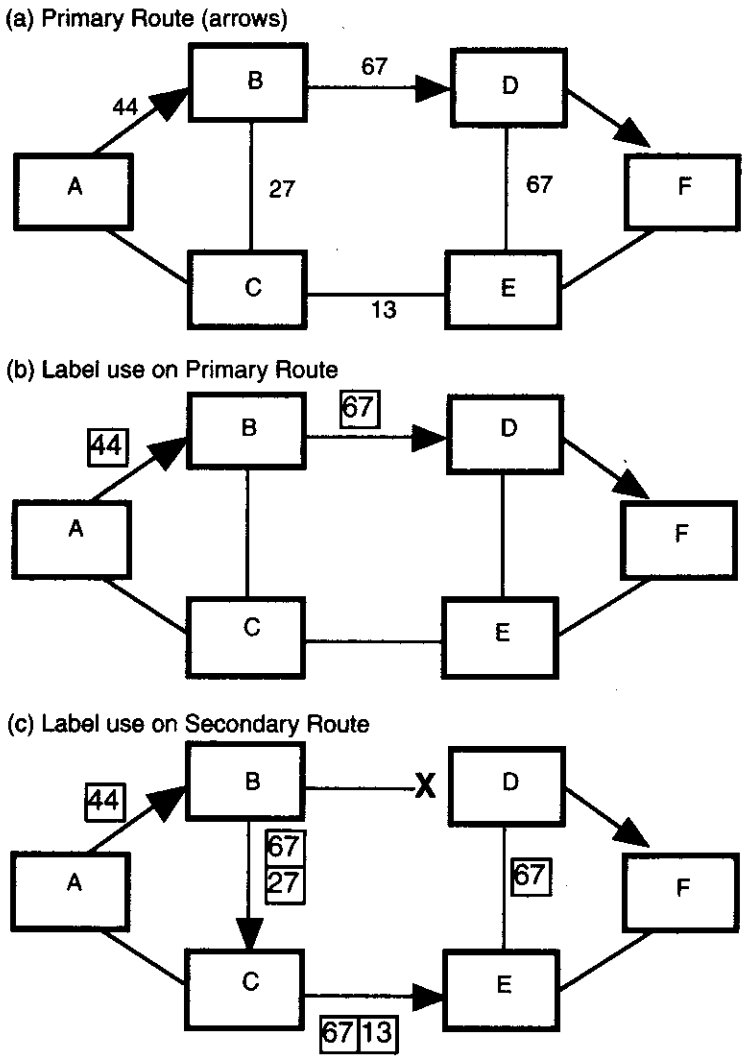


Figure 12–10 MPLS protection switching and backup routing.

In Figure 12–10 (a), the traffic is forwarded across the primary LSP from LSR A to LSR F, through LSRs B and D. The other labels shown in Figure 12–10 (a) are the labels for the backup path, and they will be explained shortly.

As shown in Figure 12–10 (b), labels 44 and 67 are used for this LSP, and at LSR D, a label pop terminates the MPLS tunnel.

In Figure 12–10 (c), the link between LSR B and D fails, or perhaps node D goes down. LSR B detects this failure (by not receiving an acknowledgment to its Hello messages from LSR D). By prior arrangement (by using, say, OSPF), LSR B knows the backup path for this LSP is to LSR C, and that the label for this part of the tunnel is 27. LSR B is configured to push label 67 into the stack behind label 27. Recall that label 67 was to be used at LSR D.

A label swap occurs at LSR C (13 for 27). Label 67 is not examined, since it is not at the top of the stack. At LSR E, label 13 is popped, leaving label 67 as the only label that arrives at LSR D. LSR D is configured to know that this label is associated with the same LSP as the one with the same label number emanating from LSR B. Thus, the LSP is protected by configuring alternate paths through the MPLS network.

PROTECTION OF THE OPTICAL SWITCHED PATH (OSP)

In order to take advantage of the speed and efficiency of optical switching, as well as the traffic engineering aspects of MPLS, it is possible to correlate the MPLS label with the optical channel, specifically a wavelength on the fiber. Using the previous example, Figure 12–11 shows how node B performs the correlation. One entry in node B's LSP cross-connect table has been provisioned for label 44 on interface c. The primary path

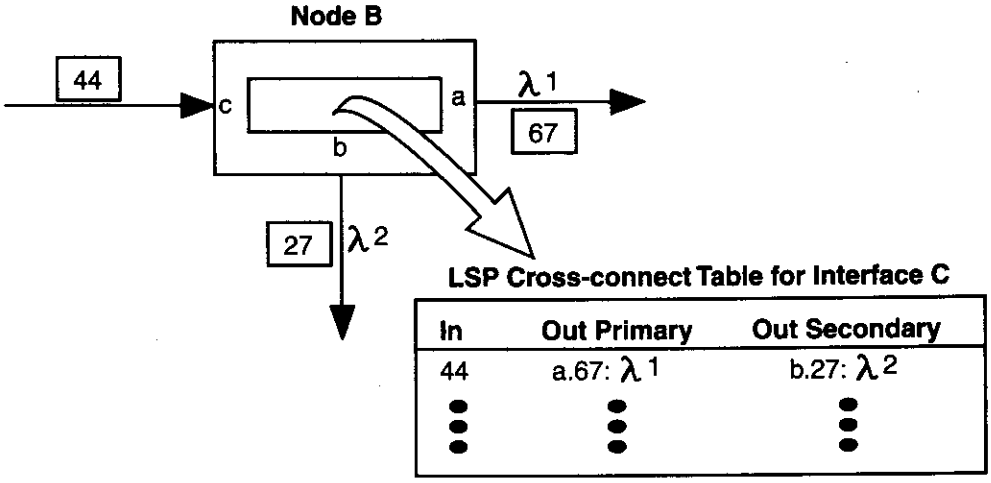


Figure 12–11 Correlating MPLS labels and optical wavelengths: The O/E/O OXC.

out of the node is through interface a. The label value of 44 is mapped to label 67 for this interface. The traffic associated with label 44 (as well as the label header of 67) is then placed on λ1 at interface a for transmission to the next node. The secondary (backup) route has also been set up, with a mapping of label 44 to label 27, and a cross-connect to λ2 on interface b. Similar cross-connect tables are configured for all the optical nodes that are part of the LSP.

The example in Figure 12–11 represents an O/E/O optical/electrical cross-connect (OXC), probably situated as an edge router to a routing domain (the ingress node to a backbone optical network). The optical signals are converted to electrical signals in order to execute hardware and software in the node for making the cross-connect and mapping decisions. Obviously, the O/E/O operations must take place at some of the nodes in the network, but it is desirable to avoid the signal conversions.

The next example in Figure 12–12 shows how a O/E/O cross-connect operating in the core (in the backbone) of the optical network would handle a relaying operation.

In this operation, node B is playing the role of an O/E/O XC but does not examine the bits on the wavelength. Instead, the wavelengths themselves provide the information to make a cross-connect switching decision. The optical switch is configured to reflect *and convert* incoming λ3

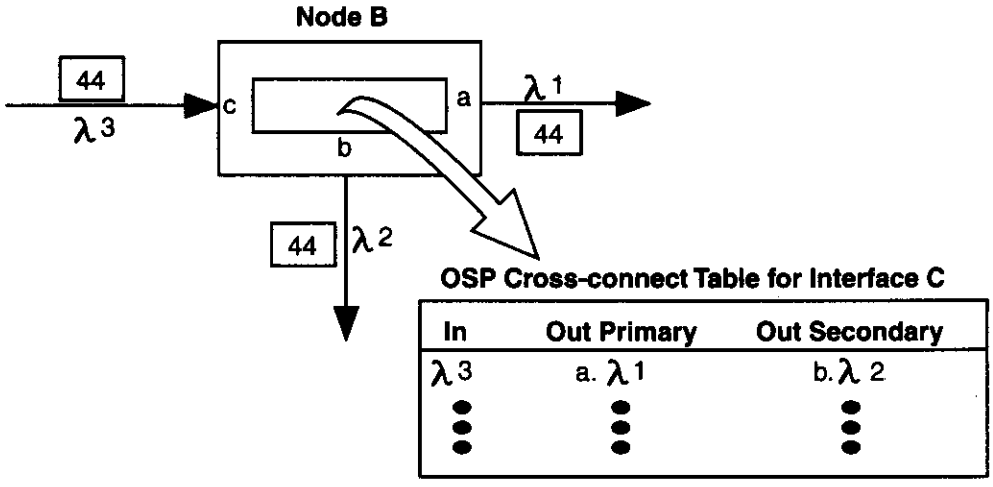


Figure 12–12 The O/E/O OXC.

on interface c to either λ1 on interface a, or λ2 on interface b. The conversion (at this stage of the practice) requires electrical intervention.

Thus, the optical cross-connect table is configured to support an optical switching path (OSP). Recall that the OSP is the optical path between two adjacent nodes.

Figure 12–13 shows the switch operating as an O/O/O node. Two aspects of this node are of interest here: (a) MPLS labels are passed transparently through the node and (b) the same wavelengths are used on the three interfaces. For this scenario, the physical signal is amplified and reflected through the amplification and switching components, perhaps with a MEMS architecture.

Obviously, this operation is the fastest of the systems examined in this section of the chapter. However, there are restrictions on how many O/O/O spans the optical signal can traverse. Recall from Chapter 3 that the network designer must be concerned with the ASE, PMD, crosstalk, and chromatic dispersion performance of the link and link nodes.

Also, Figure 12–13 shows the operations taking place with use a "table." The operations can be performed in amplifiers and MEMS components; the table is for illustrative purposes.

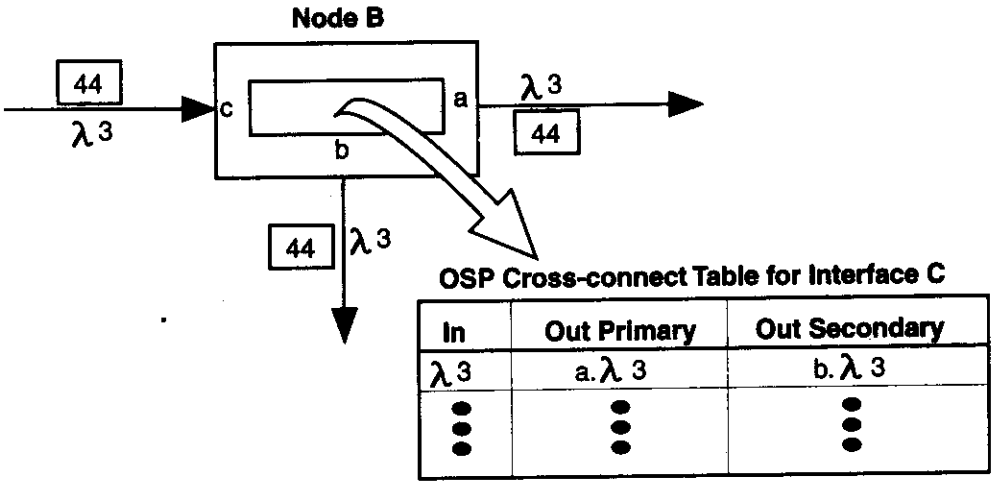


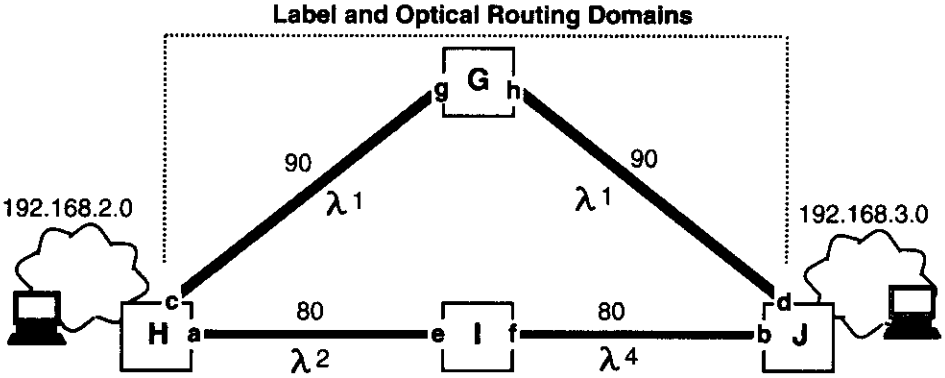
Figure 12–13 The O/O/O PXC.

CORRELATING THE WAVELENGTH OSP WITH THE MPLS LSP

Mapping a label to a wavelength at an optical node is not a difficult technical job, and photonic switching is now viewed as not only feasible, but also desirable. The task for this part of the chapter is to resolve how an optical node can exploit the traffic engineering capabilities of MPLS, as well as the fast switching capabilities of photonic switching. In explaining one approach to this issue, keep in mind that I am offering my own views on the matter, and, in some aspects, they may differ from some of my clients' views. With others we are in agreement. I welcome your views as well, since the issues in this matter are not yet settled.

Figure 12–14 illustrates a simple topology of four XCs. They are capable of both O/E/O and O/O/O operations. The central idea is to use an O/E/O control plane to set up both the wavelength OSPs and the MPLS LSPs, then execute an O/O/O data plane to relay the traffic through the interior XCs in the core part of the network for nodes that use the same λ on the end-to-end lightpath, and an O/E/O data plane for nodes using different wavelengths.

Three subchannels (wavelengths) will be set up between the four nodes, named G, H, I, and J. The wavelengths are λ1, λ2, and λ4. The three wavelengths between the nodes represent four OSPs; two of the OSPs use the same wavelength (λ1). Furthermore, two MPLS LSPs are set up between nodes H and J. Since MPLS allows label binding between



LSP 80: H and J, through I
LSP 90: H and J, through G

Figure 12–14 Optical switching paths (OSPs) and label switching paths (LSPs).

nonadjacent nodes, it is not necessary for nodes G and I to participate in the MPLS label binding operations between nodes H and J. Certainly, nodes G and I must pass the binding messages back and forth between H and J. But that is the point: Nodes G and I are pass-through nodes with regards to label distribution and binding.

Nodes H and J are designated as edge routers for this switching domain. In this role, they must be capable of executing both the MPLS and optical control and data planes. Furthermore, they must be able to map IP addresses to MPLS labels.

Nodes G and I are core routers. In this example, these nodes do not become involved in the MPLS data plane. Their main job is to perform photonic switching using the OSP cross-connect table.

Prior to the OSP and LSP operations, a routing protocol such as OSPF has been used to inform node H that IP address 192.168.3.0 can be reached through node I. When a user at network 192.168.2.0 sends traffic to the user at address 192.168.3.0, the four nodes use the data (forwarding) plane to relay the traffic. In order to accomplish this transfer, the operations take place as described in the next section.

Setting up the OSPs and LSPs Between Nodes H, I, and J

The following events can occur between nodes H, I and J to set up optical switching paths between nodes H, I, and J, and label switching paths between nodes H and J. These events do not have to occur in the exact order described here. For example, event 2 can precede event 1.

- Event 1: Node H knows it can reach node J (and 192.168.3.0) via node I because of IS-IS, OSPF, or BGP. (Note: Node H must have the complete topology map of the other nodes.)
- Event 2. Node I sets up its OSP cross-connect table to correlate (map) $\lambda 2$ on interface e and $\lambda 4$ on interface f to node H and node J, respectively.
- Event 3: Node I uses GMPLS, LMP or some other optical control plane protocol to configure $\lambda 2$ and $\lambda 4$ between nodes H and J, respectively. Consequently, node H knows it can reach node I on $\lambda 2$ at interface a. Node H does not care (or know) about the OSP relationship of interface f, $\lambda 4$ that node I has with node J.
- Event 4: Nodes H and J bind label 80 for all traffic with address 192.168.3.0 between them on their respective interfaces of a and b. This operation may begin at Node J because it is the downstream node (the next hop) to address 192.168.3.0. Node I merely acts

as a pass-through node for this label distribution and binding operation; it does not build an entry in its LSP cross-connect table. Therefore, label 80 is not processed by node I. Let's assume label 80 is associated with a certain class of service (say, delay) as well as with 192.168.3.0. Therefore, node H knows that any traffic that is destined for 192.168.3.0 can be sent out to node I on λ2 at interface a. Node J knows that a packet arriving on λ4 via interface b is intended for 192.168.3.0. Of course, node J must be O/E/O-capable, and convert the λ4 optical signal to an electrical signal so that it can examine and process the label. Likewise, node I is performing O/E/O operations as it converts λ2 to λ4.

  - Note: Node I can certainly participate at the label level and be part of the LSP. This example keeps node I away from LSP management.

- Event 5: Thereafter, when node H receives a IP datagram destined for 192.168.3.0, it knows (due to say OSPF) that this address is reachable via node I. It also knows node I is reachable via interface a on λ2. Because of event 4, it appends label 80 to the IP traffic and sends the packet to node I.
- Event 6: Node I does not care about the MPLS label. Any traffic coming in from interface e or λ2, is switched to output interface f on λ4. Thus, this core router has performed only λ data plane operations, with O/E/O operations.
- Event 7: As the edge router to this domain, node J receives the packet via interface b on λ4. It converts the optical signal to an electrical signal and examines label 80, which reveals that node J is the end of the MPLS label switching path. Therefore, it removes (pops) label 80 and passes the IP datagram to the conventional layer 3 forwarding process for IP address look-up and final forwarding to 192.168.3.0.

Setting Up a Protection Path Between Nodes H, G, and J

In order to establish a protection (backup) path between the two edge routers, nodes H and J, similar events just described take place between H and J, and from a different core router, node G:

- Event 1: Node H has a full topology map of the other nodes and knows it can also reach node J (and 192.168.3.0) via node G.
- Event 2. Node G sets up its OSP cross-connect table to correlate (map) λ1 on interface g and λ1 on interface h to nodes H and J,

respectively. Thus, this LSP can use O/O/O lambda switching end-to-end.

- Event 3: Node G uses GMPLS, LMP or some other optical control plane protocol to configure λ1 between node H and node J. Therefore, node H knows it can reach node G on λ1 at interface c. Node H does not care (or know) about the OSP relationship of λ1 that node G has with node J.
- Event 4: Nodes H and J bind label 90 for all traffic with address 192.168.3.0 between them on their respective interfaces of c and d. Node G merely acts as a pass-through node for this label distribution and binding operation; it does not build an entry in its LSP cross-connect table. Let's assume label 90 is the exact same forwarding equivalence class as label 80. Therefore, node H knows that any traffic that is destined for 192.168.3.0 can also be sent out to node G on λ1 at interface c. However, this LSP is used as a protection path. The LSP represented by label 80 is the working (primary) path. As before, node J is configured to process labels, so it can terminate the LSP tunnel and pass the IP traffic to the user.
- Event 5: Thereafter, when node H receives an IP datagram destined for 192.168.3.0, it knows (due to, say, OSPF) that this address is reachable via node G, but node I is the primary node. It will not use the protection path unless the working path fails, or unless the network provider chooses to configure the system to use the backup path of load-leveling of traffic, or other network-specific functions.

In this example, node G has assumed the role of a pure O/O/O PXC, and no wavelength conversions are performed for this LSP.

Recovery and Use of Protection Path

For the next example, node I continues its role of a lambda cross-connect. The entry in its OSP cross-connect table correlates λ2 on interface e to λ4 on interface f. Again, node I (and G as well) is not concerned with MPLS label analysis, which would require label operations.

In the event that problems occur on the interfaces/links on the primary (working) path, the following events occur:

- Event 1: Node H discovers that the OSP between node H and node I is down by not receiving a control message (say, an LMP hello message, explained in Chapter 11) from node I within a set time. Alternately, node H could fail to receive a hello for the MPLS control

plane via the Label Distribution Protocol (LMP), assuming Node I is running the MPLS control plane (which it is not in this example).

  - Note: Of course, if an alternate wavelength or fiber in the link set between nodes H and I is available, then recovery can be made without resorting to protection switching to node G.

- Event 2: Node H consults its IP forwarding table (or better, its MPLS label forwarding table), and ascertains that 192.168.3.0 can be reached via a backup through node G. Due to previous label binding operations between nodes H and J, node H knows to append label 90 to the IP packet. Of course, node H also knows that λ1 on interface c must be used for this backup LSP and OSP.
- Event 3: Node G receives the traffic. Just like node I for the primary path, node G does not care about the MPLS label. Any traffic coming in from interface g on λ1 is switched to output interface h on λ1. Unlike node I in the previous example, this core router has performed only O/O/O operations.
- Event 4: As the egress router to this domain, node J receives the packet via interface d on λ1. It converts the optical signal to an electrical signal and examines label 90, which reveals that node J is the end of the MPLS label switching path. Therefore, it removes (pops) label 90, and passes the IP datagram to the conventional layer 3 forwarding process for IP address look-up and final forwarding to 192.168.3.0.

Expanding the Roles of Nodes G and I

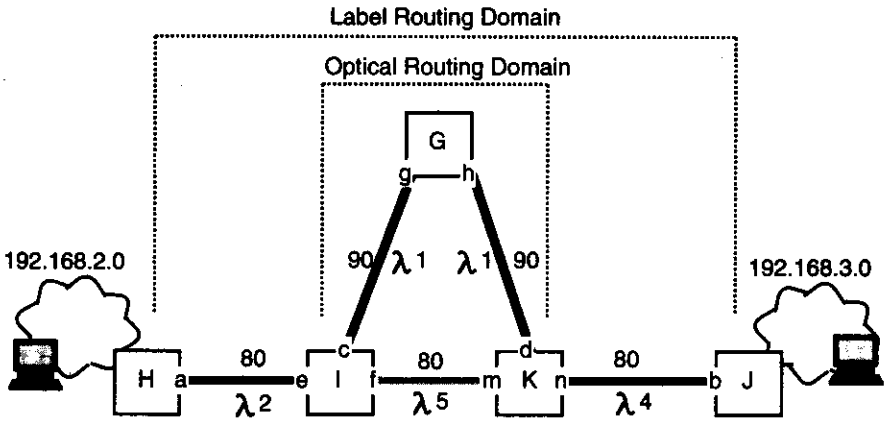
This example kept nodes G and I isolated from the MPLS operations. As noted, the nodes do not need to be so constrained. They could participate in the label binding operations with nodes H and J. Like nodes H and J in the previous examples, Nodes G and I could use an optical control channel to detect problems, and resort to label and/or optical protection switching to recover.

However, if this expanded role is implemented, nodes G and I must be able to swap labels as part of the MPLS label switching operation. Thus, the role expansion places nodes G and I back into the job of being MPLS switches. Because of this restriction, my examples have confined the core routers' role to the optical data and control planes.

Also, by restricting core nodes to the optical plane, nodes G and I have less processing overhead than the nodes at the edge of the routing domain; that is, routers H and J.

NESTING THE LSPS AND OSPS

We have discussed the idea of confining the core nodes to the optical planes. Another example should help explain this concept. Figure 12–15 shows a network topology with an additional node in the backbone, router K. The figure also shows that two routing domains exist (they also existed in Figure 12–14, but are not as well-bounded as in Figure 12–15). Nodes H and J are edge routers, and nodes I, G, and K are core routers.



Node I: for Interface e

In	Out Primary	Out Secondary
λ2	f. λ5	c. λ1
⋮	⋮	⋮

Node G: for Interface g

In	Out Primary	Out Secondary
λ1	h. λ1	Not Shown
⋮	⋮	⋮

Node K: for Interface d

In	Out Primary	Out Secondary
λ1	n. λ4	Not Shown
⋮	⋮	⋮

Figure 12–15 Nesting the routing domains.

The optical routing domain is nested within the label routing domain. The nesting approach means that nodes G, I, and K are not involved in label operations for ongoing payload, as suggested by the optical cross-connect tables shown at the bottom of Figure 12–15. It is the job of nodes H and J to correlate labels to the wavelengths associated with the optical routing domain. In this manner, the operations proceed as in previous examples, except that the error recovery is moved to the optical domain, and nodes H and J are no longer involved in error detection.

The relationships are as follows: (a) LSP 80 uses O/E/O for the OSPs at nodes I and K, and (b) LSP 90 uses O/O/O for the OSPs at node G.

There is one remaining problem with this scenario. The example in Figure 12–15 shows only one path to/from the optical backbone for nodes H and J. Consequently, if the links from these nodes to the network fail, the customers at nodes H and J are denied service. If up time and robustness are of paramount importance, then nodes H and J must be outfitted with alternate links to other core nodes, say, node G. This arrangement puts us back to the topology in Figure 12–14. The important point here is that the users sitting behind nodes H and J may not need 100% availability, and they may be willing to accept occasional downtime for a less-expensive configuration

To conclude the discussion of label and optical switching paths, let's assume the link between nodes I and K fails in Figure 12–15, and therefore, $\lambda 5$ is not available to convey the user payload between these two nodes. The following events take place to recover from this failure:

- Event 1: Node I does not receive acknowledgments to its LMP hello messages that it sent to node K on the optical control channel on interface f.
- Event 2: Node I times-out on the LMP hello procedure with node K, and consults its optical cross-connect table for a protection path.
- Event 3: Node I determines that the secondary (protection) path is $\lambda 1$ on interface c. It reconfigures the MEMS mirrors to (literally) reflect this observation. The traffic is diverted to node G. This operation also requires OEO $\lambda 2$ to $\lambda 1$ conversion.
- Event 3: Node G performs its ongoing optical cross-connect O/O/O operation.
- Event 4: So does node K with O/E/O operations, which also results in the traffic reaching node J, where the operations previously explained take place.

TOPOLOGY CHOICES FOR A NODE FAILURE

Thus far, the examples have focused on failures of optical links, with the optical nodes still up and running. In the event of failure of a node, the recovery scenario is more complex, and more expensive to implement. Nonetheless, it might be necessary, in the optical network core, to use BLSRs, BPSRs, or semi-meshed topologies discussed in Chapter 8. Whatever the topology may be, it is still desirable to keep the core nodes isolated from the upper layer MPLS label operations.

PLANE COUPLING AND DE-COUPLING

Notice that my explanations of the planes' interworkings show that the three data planes (IP, MPLS, and λ) are not coupled together. That is, a router can perform forwarding operations in a data plane without regard to the operations occurring in another data plane. But the control planes are coupled together.

As an example, an incoming IP datagram can be forwarded using the conventional IP routing table created by OSPF, IS-IS, or BGP. Alternatively, the datagram's destination address prefix can be mapped to an MPLS label by the IP and MPLS control planes' interactions. But thereafter, the labeled packet is processed only by the MPLS data plane. Likewise, the MPLS and λ control planes can interact to set up a relationship between labels and the OSP cross-connect table. Thereafter, wavelengths are processed only by the optical control plane, and the MPLS and IP headers are not examined.

Notice also that control plane interworking is needed to set up LSP and OSP protection paths. But once the protection paths are configured, protection switching can be performed at the optical level, without executing MPLS or IP operations.

Whether or not this approach will work is dependent upon how frequently control planes must be invoked. If LSPs and OSPs must be set up and torn down frequently, the resulting overhead will affect overall network performance. But this situation is a problem in any network that provides dynamic services and adaptive responses to customer requirements. Even with frequent control plane instantiations, keeping the core/backbone nodes focused on lambda switching and pushing IP/MPLS/λ control plane interactions to the edge routers should help considerably in ameliorating the overhead problem.

SOME END-TO-END WAVELENGTHS AND SOME NODE-TO-NODE WAVELENGTHS

A reasonable approach in implementing MPLS and optical planes is to try to set up end-to-end LSPs using the same wavelength. I am certain this possibility exists in large networks with large user populations; it is just like a leased line circuit, with the digital cross-connect fabric set up and remaining static. For those applications that do not fit a fixed, static profile, GMPLS can be used to try to allocate compatible wavelengths, and, if unsuccessful, to confine the core switches to a fast lambda conversion routine, but not label processing.

Some papers state that (eventually) wavelength conversion performed with tunable devices will perform at sufficient speeds to make the issue of wavelength conversion a moot point. Perhaps so, but in the meantime, a conservative and prudent apprach to MPLS label and lambda interworking is a cornerstone to an effective 3rd generation optical Internet.

GRANULARITY OF LABELS VS. WAVELENGTH SUPPORT

Regardless of the pros and cons of building systems that integrate the features of MPLS with the optical capabilities, the issue of label granularity must be carefully weighed against the number of wavelengths to support/transport each label.

To explain this important point, consider Figure 12–16. Node H is connected to node I with one fiber link set, and node I is connected to nodes G and K with one fiber link set to each node. Node H needs to send labeled packets to both nodes G and K. If node I is to perform photonic switching, node H cannot send MPLS packets destined for nodes G and H to node I on the same wavelength. If this situation does occur, node I must resort to O/E/O operations, convert the optical signal to an electrical signal, interpret the label value, make a switching decision, and convert the electrical signal back to an optical signal for transport to either node G or K.

Therefore, for an optical internet that operates with all three control planes (IP, MPLS, and λ) to be effective, the following situations should exist:

- There should be an ability to aggregate many IP address prefixes at the ingress node for O/O/O transport to the egress node. That is, the LSP and the associated wavelength should not change through the backbone network (and with some exceptions, the user-to-user LSP tunnel should extend through the network anyway).

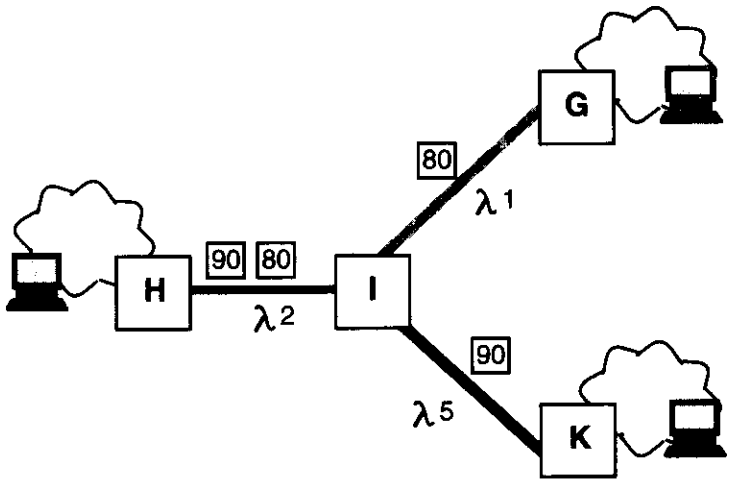


Figure 12–16 Label and wavelength granularity.

- Core nodes in the backbone can execute the O/E/O control planes to set up and correlate TE MPLS LSPs with the OSP cross-connect tables, and thereafter use the O/O/O data plane for ongoing traffic transfer.
- The interior nodes can use O/O/O-oriented λ protection switching, as described in this chapter, or, in the event of a problem, a node can resort to O/E/O control plane operations to reconstruct both the LSP and contributing OSPs.
- If the optical nodes and links are of high quality (and downtime is rare), the frequency of executions of the O/E/O-oriented control planes to allow core nodes to recover from occasional problems should not place an undue burden on the resources of the network.
- Notwithstanding these points, IP address aggregation is important to both the LSPs and the OSPs.

APPROACH TO THE PROBLEM OF LSP AND OSP INTERWORKING

In conclusion, at this early stage in the evolution of 3rd generation transport networks, one of the big challenges is to be able to keep the backbone confined to O/O/O operations most of the time, push the O/E/O operations of IP, MPLS, λ mapping to the ingress and egress nodes, and try to aggregate as many IP address prefixes into as few labels as possible.

Opponents of this approach will surely state that this gross aggregation approach will work against the important issue of supporting (and charging for) tailored QOS for each user (or a small subset of a community of users). After all, aggregating multiple users to one label with different QOS needs defeats one of the goals of a 3rd generation transport network: providing tailored QOS for individual users.

I do not hold this view. When the optical network has evolved to a multi-terabit transport system, the core network's performance will be such a high capacity that a node will not have the time to tailor its behavior to meet each user's performance requirements. Nor should it have to do so; the network will be operating at such a high bit rate and providing such low delay, that the issue of QOS support will be addressed only at the edge nodes of the network, those that provide ingress and egress for the customers' traffic.

Then why even bother with QOS provisioning and monitoring at the user-network interface (UNI) to and from the edge nodes? If the network is so all-powerful, why not just eliminate the time-consuming QOS and traffic policing functions?

The answer is two-fold. First, QOS and traffic policing remain essential in order for the network provider to charge for its services, and to regulate traffic at the (probable) bandwidth-constrained UNI. Second, the idea of a transport network that has all the capacity it needs is still just that: an idea.

From the standpoint of the optical fiber itself: Yes, it is known that enough cable can be laid to support any foreseeable demand. The ultimate challenge is to also design the optical nodes *and* the many servers and routers in the backbone to operate at a capacity in consonance with optical fiber. We return to this critical issue at the end of this book.

MEMS AND OPTICAL SWITCHING RE-EXAMINED

The optical switch technology is far from mature, and the interworking of IP, MPLS, and wavelengths is in its infancy. But, as noted in the preface to this book, we are looking forward to what might evolve in 3G transport networks and optical Internets. For all these wonderful IP, MPLS, and wavelength networks to come into being, MEMS and other optical switching technologies must be improved considerably, both in speed and reliability. But most people in the industry think the scenarios described in this chapter will be in existence in a few years. We must wait and see.

For a review of the state of optical switches and MEMS, I recommend [FULL01] as follow-up reading.

THERMO-OPTIC SWITCHES

There are other optical switches under development. The MEMS has gained favor and therefore has been highlighted in this chapter. However, as of this writing, it is not clear whether or not the optical switches are going to need switching matrices that support thousands of ports. It appears the MEMS technology is being pushed into the future, and very large photonic cross-connects have not reached a point of high demand.

Anyway, we shall wait to see what happens in the marketplace. For now, another optical switch that is being developed is the thermo-optical switch, also called a photonic waveguide. It is also built from silicon, and, like MEMS, it is very small. Its operation is shown in Figure 12–17. The light wave enters the switch at a splitter, which splits the beam and sends both copies down the waveguides. One of the pipes has its temperature changed by heating an electrical resistor. This heating changes the length of the waveguide, which results in the changing of the phase of the light passing through it. The two wavelengths reconverge at a coupler, which is capable of switching (or not switching) the waveguides onto an output port. In the example in Figure 12–17, wavelength 2 is switched onto output port 2, but wavelength 1 is not passed out of the switch.

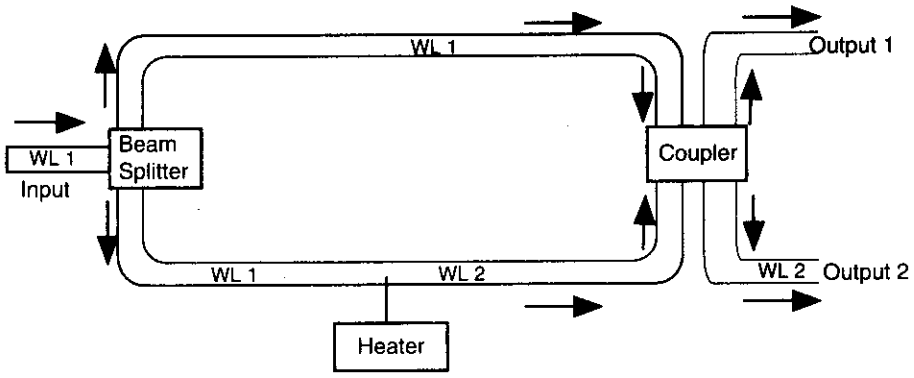


Figure 12–17 Thermo-optic switches.

Bubble Switches

Another optical switch fabric is the bubble switch, which is a very small bubble of fluid whose refraction index is the same as silica. This example also uses inkjet printer technology and is being developed by Agilent. The switching of the lightwave takes place if a bubble is in a fill hole. The fill hole can be filled with the refraction material dynamically, causing the light to bend and be transferred to another waveguide. If the fill hole is empty, the light continues through the original waveguide.

SUMMARY

The emerging third generation transport network will be capable of photonic switching, although O/O/O nodes are just now in the stages of development. The 3G transport systems will interwork the IP, MPLS, and λ control planes. The scenarios presented in this chapter represent one method of control plane interworking to support network switching requirements, and I expect the IETF working groups will soon publish their views on this important operation.

Whatever the specific method chosen, MPλS will require interactions between the MPLS and λ control planes. It is the data planes that should remain as loosely coupled as possible. I have presented such a scenario in this chapter.

13

ASON Operations at the User Network Interface (UNI) and the Network-to-Network Interface (NNI)

This chapter describes the emerging Internet specifications on the automatic switched optical network (ASON) operations at the UNI and NNI. These are important blueprints, for they establish rules for the interactions of user nodes, edge nodes, and interior nodes as they communicate with each other. The focus in this chapter is three-fold, and, of course, is related to information in other chapters: (a) managing bandwidth and providing bandwidth on demand, (b) rules for UNI interactions, and (c) rules for NNI interactions.

The material in this chapter is also related to some topics in Chapter 10 (see "Two General Models"). Also, be aware that NNI also means Network Node Interface.

OBJECTIVES OF THE ASON

The IETF has defined the following objectives for an ASON:

- Use of a standardized optical network control plane in order to achieve interoperability among the different vendor products.
- Ability to support rapid and automatic end-to-end provisioning of services within the network.

- On an individual service path basis, provide (if necessary) dynamic routing, restoration, usage-based accounting, policy-based quality of service features, and billing based on services rendered.
- Insofar as possible, keep one vendor's upgrade (vendor A) of the control plane transparent to another vendor (vendor B). This objective means that a control plane version upgrade should not have to be patched to all vendor systems in a multivendor network in order for vendor A's customers to obtain the benefit of the upgrade.
- Provide adequate security protection of the optical layer, particularly the control plane. Adequate security assumes that the service path can traverse multiple networks (e.g., different network administrations [domains]), and across NNIs, and receive proper and correct security services end to end. So, a domain assumes that its traffic is safe when the traffic traverses another domain.

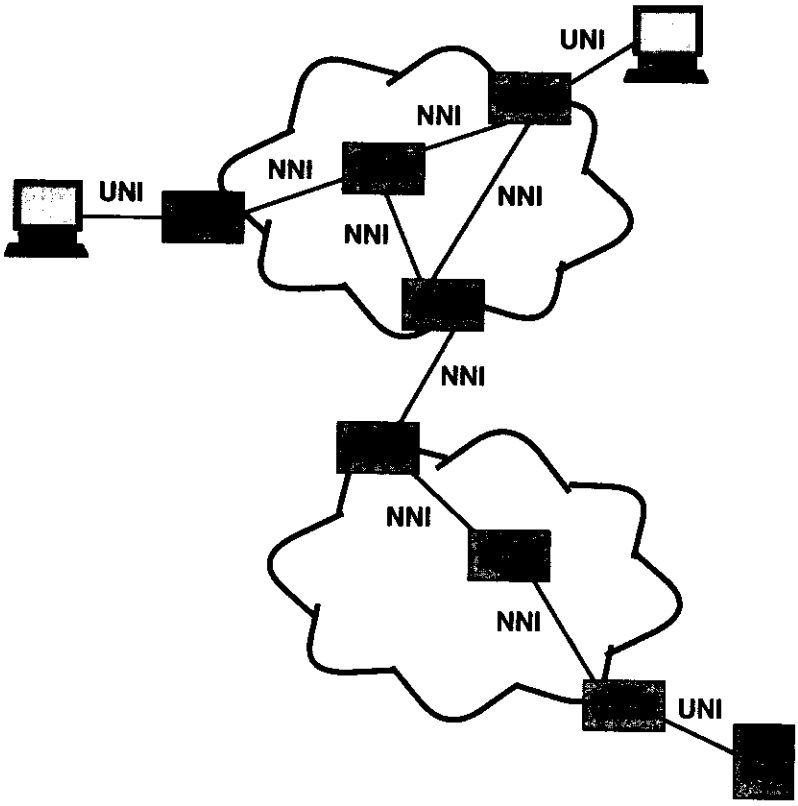


Figure 13–1 The UNI and the NNI.

- Allow the network service provider to control the usage of its network resources, giving the provider the ability to ensure mission-critical service paths (perhaps for high-paying customers) which are given priority over less important service paths.
- Emphasize the user of standards and open protocols.
- Ensure the scalability of the network.

THE UNI AND THE NNI

Logically enough, the UNI defines the operations between the user node and the ingress and egress network nodes, as seen in Figure 13–1. The NNI defines the operations between network nodes between networks, as well as within networks.

These two interfaces differ in some ways, and in others they are the same or identical. The services defined at the UNI are (a) neighbor discovery, (b) service discovery, and (c) signaling operations to create, manage, and modify an optical connection. The services defined for the NNI are (a) neighbor discovery, (b) topology and resource distribution (based on OSPF), (c) traffic engineering, and (d) signaling (based on CR-LDP or RSVP-TE).

MANAGING THE OPTICAL BANDWIDTH IN THE ASON

Bandwidth on demand has been provided by network operators for many years. Examples are the switched virtual circuits (SVCs) of X.25, Frame Relay, and ATM. The plain old dial-up services are another example (somewhat limited in its capabilities) of bandwidth on demand.

With the development of optical networks that provide tremendous bandwidth capacity, there is keen interest among customers, vendors, and network providers to deploy network offerings that provide optical bandwidth on demand. But the issue goes further than offering bandwidth on demand. The bandwidth capacity of optical networks is increasing each year by many orders of magnitude, and a pressing issue is to develop mechanisms to manage and control this vital network resource.

The present 1st and 2nd generation transport networks are preconfigured to provide bandwidth with proprietary element management systems (EMS), and they require crafting operations before the bandwidth is made available to the customer. Due to some manual operations in-

volved in this crafting, and also because end-to-end provisioning often occurs through vendor-specific systems, it is not unusual that a long provisioning time is required before the service is made available to the customer. This situation occurs when the customer demands short provisioning times.

One idea of 3G transport systems is to migrate to a scheme in which the end user can dynamically request bandwidth from the network, and the network can dynamically find and reserve the required bandwidth for this user. Thereafter, the bandwidth is guaranteed, something like a leased, point-to-point circuit.

Inherent in optical bandwidth on demand is the ability of the user to (a) request a new connection to the network, (b) request a tear-down of an existing connection, (c) query the network about the characteristics of an ongoing connection, and (d) change parameters (and thus operating characteristics) of an ongoing connection.

Several efforts are underway to define standards for the management of bandwidth in optical networks. One effort is sponsored by the Open Domain Service Interconnect Coalition [ODSI01], and the other is sponsored by the Optical Interworking Forum [OIF01]. The IETF is taking the lead in the Internet standards area.[1] This next section provides a summary of the protocols and interfaces being defined by these two organizations and the IETF.

THE GENERAL APPROACH TO OPTICAL BANDWIDTH MANAGEMENT

Figure 13–2 shows an optical backbone in which three optical nodes (ON) are connected to the network users. The inside of the network has spare bandwidth available for its users. When a user needs some bandwidth, it sends a bandwidth request message to its local ON across a standardized UNI. The request message contains sufficient information for the network to know (a) the identification of the sending user, (b) the identification of the endpoint to which the sending user wants to send/receive traffic, and (c) the amount of bandwidth required for the connection.

[1]The best way to obtain up-to-date information on the working groups and their papers on optical networks is to go to www.ietf.org, click on working papers (or working groups), or key in a search for (a) optical and then (b) MPLS. These two subjects are closely associated with each other.

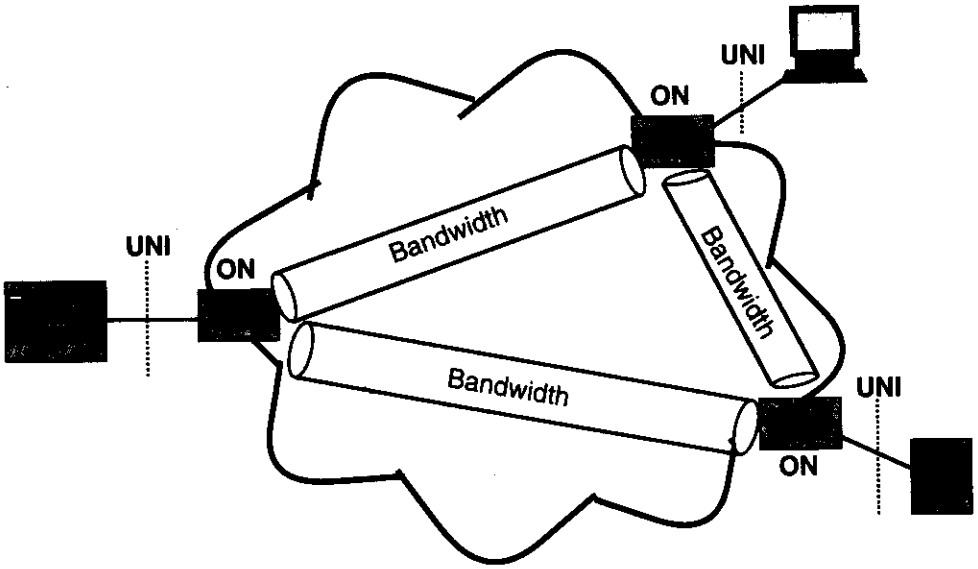


Figure 13–2 The optical bandwidth management arrangement.

Sound familiar? It should, because it is identical to the operations involved in an ATM, Frame Relay, or X.25 SVC connection on demand. A logical question to ask is why not create revisions to these currently-existing protocols since they are already available. The answer is that these emerging optical UNI specifications do not rely on conventional switched virtual circuit technologies. As we shall see, they use either the SONET overhead bytes (the DCC bytes in the section and line headers), or they use the resource reservation protocol (RSVP).

IETF OPTICAL CARRIER FRAMEWORK FOR THE UNI

The IETF is working on a model and framework for an optical carrier network [XUE01]. The general specification defines the carrier (network operator) service requirements for automatic switched optical networks (ASON). It guides the on-going efforts to develop a standard UNI and other interfaces to the optical layer (OL) control plane. It is a blueprint that describes the optical transport services for the UNI and is also based on the efforts of the OIF.

Focus on OC-48/STM-16 and Above

The primary focus is on the SONET/SDH tributaries of OC-48/STM-16 and above. While sub-rate extensions are discussed in a general way, it is important to emphasize that the philosophy, due to the nature of wavelength switching in an optical network, it is not practical to provision and switch user sub-rate signals (called user service paths) within an all-optical network. Therefore, it is necessary to include an optional low-granularity grooming optical networking function in the UNI architecture, called sub-rate UNI (UNI-SR). This interface extends to the user device and is defined as an extension to the UNI.

Table 13–1 Connection Granularity [XUE01]

SDH Name	SONET Name	Transported Signal
RS64	STS-192 Section	STM-64 (STS-192) signal without termination of any overhead (OH)
RS16	STS-192 Section	STM-16 (STS-48) signal without termination of any OH
MS16	STS-48 Line	STM-16 (STS-48); termination of RSOH possible
MS64	STS-192 Line	STM-64 (STS-192); termination of RSOH
VC-4-64c	STS-192c-SPE	VC-4-64c (STS-192c-SPE); termination of RSOH, MSOH and VC-4-64c TCM OH possible
VC-4-16c	STS48c-SPE	VC-4-16c (STS-192c-SPE); termination of RSOH, MSOH, and VC-4-16c TCM OH possible
VC-4-4c	STS-12c-SPE	VC-4-4c (STS-12c-SPE); termination of RSOH, MSOH, and VC-4-4c TCM OH possible
VC-4	STS-3c-SPE	VC-4 (STS-3c-SPE); termination of RSOH, MSOH, and VC-4 TCM OH possible.
VC-3	STS-1-SPE	VC-3 (STS-1-SPE); termination of RSOH, MSOH, and VC-3 TCM OH possible
VC-2	VC-2	VC-2 (VT6-SPE); termination of RSOH, MSOH, higher order VC-3/4 (STS-1-SPE) OH and VC-2 TCM OH possible
VC-12	VT2-SPE	VC-12 (VT-SPE); termination of RSOH, MSOH, higher order VC-3/4 (STS-1-SPE) OH and VC-12 TCM OH possible
VC-11	VT1.5-SPE	VC-11 (VT1.5-SPE); termination of RSOH, MSOH, higher order VC-3/4 (STS-1-SPE) OH and VC-11 TCM OH possible

UNI-SR (Subrates)

[XUE01] recommends the UNI-SR support the sub-rates listed in Table 13–1. Each subrate is referred to as an instance of connection granularity. If you are unfamiliar with the terms in Table 13–1, take a look at Chapters 5 and 6.

TYPES OF CONNECTIONS

As shown in Figure 13–3, three network connections are defined for the model: (a) optical channels, (b) optical paths, and (c) service paths. The optical channel defines an end-to-end physical connection between two termination points such as an ADM or an XC. Since it is defined as end-to-end through the network, it implies the concatenation of one or more optical fiber links or optical wavelength channels.

The optical path is the logical connection over the optical channel through the optical network. A good way to think of the optical path is that it is the optical frame (say SONET/SDH) running on the optical signal. The optical path does not extend past the edges of the network.

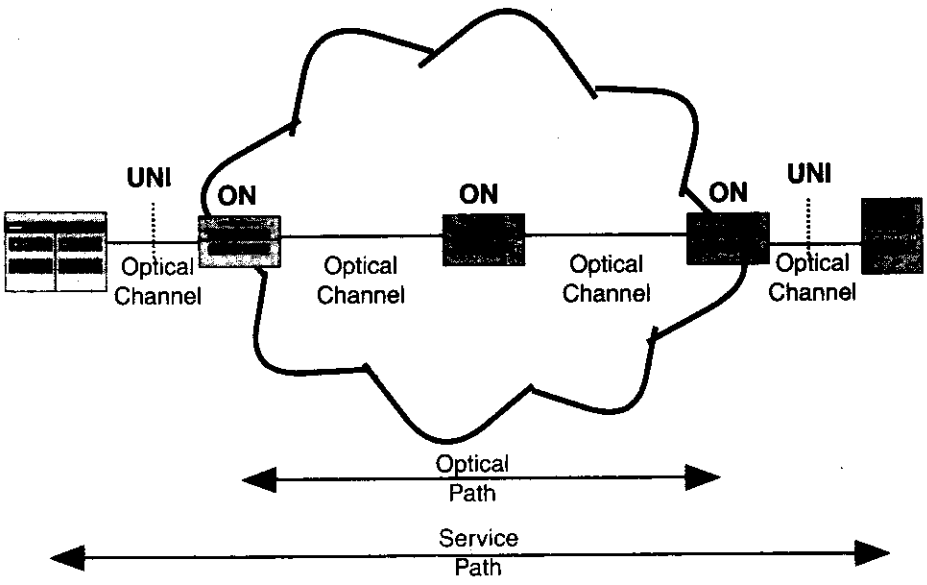


Figure 13–3 The network connections.

The service path is the logical end-to-end connection over an optical path. But "end-to-end" in this context means that the service path extends out to the user node (called the user edge device, or UED).

Connection Attributes

Some of the important aspects of optical bandwidth on demand, as defined in [XUE01], are the attributes associated with the connection. These attributes can be negotiated between the user and the network before the connection is established, or some of them can be modified during the connection. They are categorized as (a) identification attributes, (b) connection characteristic attributes, and (c) routing constraints attributes.

Identification Attributes

These attributes are used in the connection establishment operation and in the ongoing management of this connection. They include:

- Connection id: A globally unique identifier for the connection. This identifier is assigned by the network.
- Contract id: An identifier of who "owns" the connection. It is important for connection acceptance, billing, and appropriate support for SLAs, etc.
- User Group id: Identifier specifying groups of users that are associated in a group.
- Connection Status: Describes the state of the connection.
- Connection Schedule: The date/time when the connection is desired to be in service, and the earliest and latest date/time when the connection will be disconnected.
- Destination Name: The name used to identify the node to which the connection is to be established.
- Destination Port Index: If individual ports are not given unique addresses, then a port index is required to identify them.
- Destination Channel (wavelength) id: When the network or end system allows multiplexing or switching at a finer granularity below the port level, the channel identifier is used to refer to specific channels below the port level. The destination channel identifier may be assigned by either the destination user or the network.
- Destination Sub-channel id: A further level of destination multiplexing.

- Source Name: The name used to identify the node from which the connection is to be established.
- Source Port Index: Similar to destination port index.
- Source Channel (wavelength) id: Similar to destination channel identifier.
- Source Sub-channel id: Similar to destination sub-channel ID.
- Third Party (Proxy) Attributes: Third party or proxy signaling identifies an entity other than the equipment directly connected to the interface.

Connection Characteristic Attributes

The next group of attributes relates to the physical and transmission characteristics of a connection. They are as follows:

- Framing: Designation of framing types such as: (a) SONET T1.105, (b) G.707, (c) Ethernet IEEE 802.3.x, (d) Digital wrapper G.709, (e) PDH, and (f) transparent (wavelength service).
- OH Termination Type: This field is framing-specific. For SONET and SDH framing, this field specifies to what degree the framing overhead bytes are terminated: (a) RS: signal without termination of any OH, (b) MS: signal with termination of RSOH (section OH) possible, and (c) VC: signal with termination of RSOH, MSOH, and TCM OH possible.
- Bandwidth: This attribute is also correlated to framing type. Its values will be consistent with the allowable values within the selected framing type. For SONET, this field will be used to indicate the bandwidth of the connection in terms of multiples of STS-1 (and VTx, if applicable), to allow for virtual concatenation. For SDH, this field will be used to indicate the bandwidth of the connection in terms of multiples of VC-4s/STM-1s (and VC-3, VC-12, and VC-11, if applicable), and should allow for virtual concatenation. For Ethernet, the values are in multiples of Mbit/s, so that Gigabit Ethernet is represented as 1000, and10GbE is represented as 10000.
- Directionality: This attribute will indicate whether the connection is uni-directional or bi-directional. SONET, SDH, and Ethernet are defined as bi-directional signals.
- Protection and restoration: Priority, protection, and restoration will be represented by two attributes:

- Service type: Specifies a class of service. A carrier may specify a range of different classes of service with predefined characteristics (e.g., restoration plans). The pre-defined service types correspond to different types of restoration (e.g., no restoration, 1+1 protection), connection set-up and hold priorities, reversion strategies for the connection after failures have been repaired, and retention strategies.
- Drop side protection: Refers to the protection between the user network elements and the optical network. Two different fields will be used to specify protection used at the two ends of the connection (i.e., different protection schemes can be used at both ends of the connection).

Routing Constraints Attributes

Various relationships may be defined between connections. For example, a user may request that multiple connections be diversely routed, that multiple connections be routed along the same shared physical route, that multiple connections be bundled together and treated as a single entity with a common set of attributes, or that a given connection be routed on the same path as an existing connection.

Two connections are diverse if they have no shared risk link groups (SRLG) in common. Diverse routing is frequently a requirement of sophisticated enterprise networks whose availability objectives may require that no single failure isolate a node or disconnect the network. The diversity requirement for such a network is best expressed by a matrix: If there are N connections involved between the same end points, then A[j,k] specifies whether connections j and k must be diverse.

Connections may initially be requested with the intention of adding a diversely routed connection at a future point in time. This allows requests to be individually handled while still providing diversity at a later date without service interruption.

As of this writing, the specifics of the routing attributes have not yet been defined.

THE NETWORK-TO-NETWORK INTERFACE (NNI)

The NNI is under development by the IETF Network Working Group [PAPA00]. The NNI model is shown in Figure 13–4. Several nodes and entities are shown in this figure that have not yet been explained. They are:

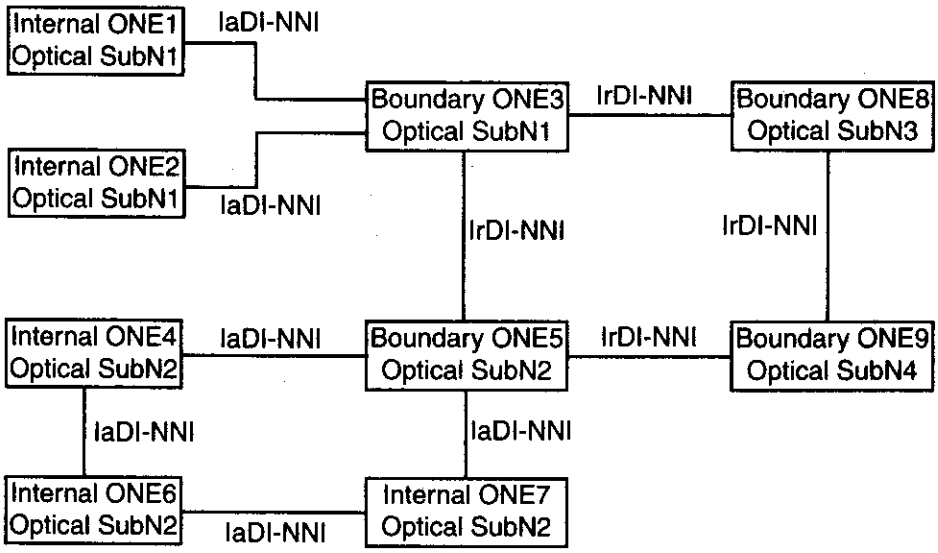


Figure 13–4 The NNI model [PAPA00].

- Optical network element (ONE): A network element belonging to the optical network. An optical network device could be an optical cross-connect (OXC), an optical ADM (OADM), etc.
- ONE controller: The owner of the UNI-N interface (since the UNI-N may not belong to the same device as the ONE) toward the UNI-C interface and/or the owner of the NNI interface.
- Boundary ONE: An optical network element belonging to the optical network whose controller includes an IrDI-NNI interface and an IaDI-NNI interface and/or a UNI-N interface.
- Internal ONE: An optical network element internal to the optical network (also referred as a termination incapable device) whose controller has only an IaDI-NNI interface.
- Client network element (CNE): A network element belonging to the client network. A client (C, that is, the customer) network element could be a SONET/SDH ADM, a SONET/SDH cross-connect, an ATM switch, an Ethernet switch, an IP router, etc.
- CNE controller: The owner of the UNI-C interface (since the UNI-C may not belong to the same device as the boundary CNE).
- Optical network controller (ONC): Logical entity within an optical sub-network terminating the NNI signaling.

- Intra-domain (IaDI)-NNI interface: The interface between internal ONE controllers belonging to the same optical sub-network or between internal ONE controllers belonging to distinct optical sub-networks.
- Inter-domain (IrDI)-NNI interface: The interface between boundary ONE controllers belonging to distinct optical networks.
- Generalized label switched path (GLSP): Point-to-point connection with specified attributes established between two termination points in the optical network. GLSPs are considered as bi-directional (and in a first phase as symmetric). A GLSP could be a fiber switched path, a lambda switched path or a TDM switched path (circuit).

In Figure 13–4, there are four optical subnetworks, four boundary ONEs, and five internal ONEs with the following relationships:

- Optical sub-network1 includes 2 internal ONEs and 1 boundary ONE
- Optical sub-network2 includes 3 internal ONEs and 1 boundary ONE
- Optical sub-network3 includes 1 boundary ONE
- Optical sub-network4 includes 1 boundary ONE

NNI Signaling Requirements

The following signaling transport mechanisms are defined for the NNI:

- In-band: Signaling messages are carried over a control-channel embedded in the logical link between the NNI interfaces of the peering ONE controllers. The control-channel is implemented through the use of optical channel layer (OCh) overhead bytes [G.70901] over which the NNI signaling channel is realized. For the SONST/SDH particular case, the control-channel could be implemented through line DCC bytes or other SONET/ SDH unused overhead bytes.
- Out-of-band: Signaling messages are carried over a control-channel embedded in the physical link between the NNI interfaces of the peering ONE controllers. The control-channel is implemented through the use of a dedicated wavelength included on a

(D)WDM fiber link over which the NNI signaling channel is realized. This channel is referenced as the optical supervisory channel (OSC). For the SONET/SDH particular case, a TDM sub-channel can be allocated for realizing the NNI signaling channel.

- Out-of-network: Signaling messages are carried over a dedicated and separated network between NNI agent interfaces of the peering ONC controllers or over a dedicated control-link between NNI interfaces of the peering ONE controllers. The dedicated physical-link is implemented through the use of one (or multiple) dedicated interface(s) over which the NNI signaling channel is realized.

Neighbor Discovery

The key objective of the neighbor discovery protocol (NDP) at the NNI is to provide the information needed to determine the neighbor identity (IPv4 address associated to the corresponding NNI) and neighbor connectivity over each link connecting internal ONEs or an internal ONE to a boundary ONE. The physical port and identity discovery provide the following information to the ONE:

- The ONE discovers the identity of the neighboring ONE by automatically discovering the IPv4 address assigned to the NNI interface.
- The ONE discovers the identity of the physical ports of each port connected to the neighboring ONE.

NNI Topology and Resource Distribution Protocol

The Topology and Resource Distribution Protocol (TRDP) is the mechanism provided to initially exchange and distribute the discovered logical-port-related information of the ONE included in an optical subnetwork. This protocol runs across intra-domain NNI interfaces.

The TRDP protocol is an IGP protocol and is based on the following concepts:

- Maintaining the neighbor relationship with peering ONEs
- Flooding of the ONEs' logical link adjacencies
- Flooding of the ONEs' logical link state
- Flooding of the ONEs' logical link related information

NNI Protocol Mechanisms

We must move on in this chapter to study the ASON UNI and NNI signaling services. But before we do, note that this discussion on the NNI has been a highlight of the specification as defined in [PAPA00]. If you need the details for which to make detailed planning and general design decisions, you should consult this reference.

UNI AND NNI SIGNALING SERVICES

The UNI and NNI define five and four signaling services, respectively. The services are invoked by a UNI or NNI node sending a request message to another node or nodes, as shown in Figure 13–5. In turn, the receiving node must respond with a response message. Below is a summary of the functions of these services, and Table 13–2 will be helpful as you read about the services.

- Creation: This procedure is used to create a lightpath (end-to-end) from the source UNI, through one or more NNIs, to the destination UNI. The principal services achieved are a route calculation and determination, assignment of identifiers, and the allocation of resources. Table 13–2 lists and explains the parameters that are

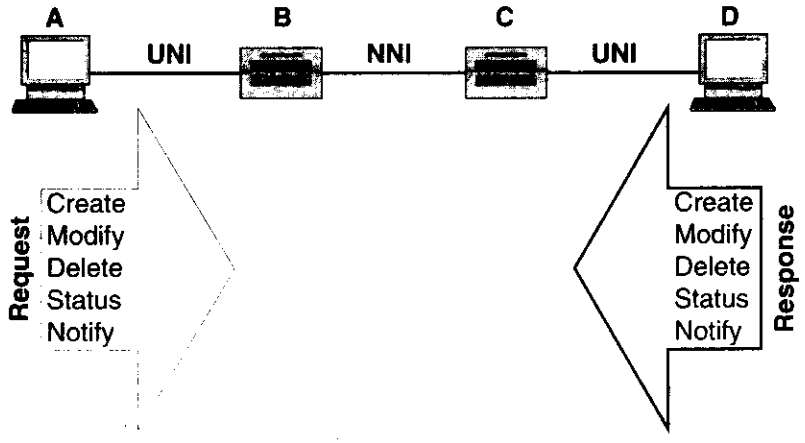


Figure 13–5 The UNI and NNI signaling services and messages.

Table 13–2 UNI and NNI Message Parameters

Parameter Name(s)	Parameter Function
Bandwidth-framing	Values will be consistent with the allowable values within the selected framing type, such as Ethernet, SONET, SDH, ATM, IP, etc.
Carrier ID	ID of carrier that created connection request
Contract ID	Owner of the connection
Directionality	Indication if connection is uni-directional or bi-directional
Diversity	Possible needs for lightpaths to traverse different conduits, etc.
Explicit route	Information on the constrained route through the optical network(s)
Max signaling delay	Maximum delay in executing signaling procedures to establish a connection
Network protection	Degree and type of NNI protection switching (1:1, etc.)
Priority-preemption	Under study, but will allow the preemption of certain GLSP processes
Record route	Information on the route taken by the GLSP
Result code	Diagnostic information
SDH/SONET options	Specific to SDH/SONET OAM procedures
Side protection	Refers to the protection between the user network elements and the optical network (1:1, 1+1, etc.)
Termination ID	ID of an optical channel that defines an end-to-end physical connection between two termination points in the network by concatenating one or more optical links or optical wavelength channels
User group ID	Identifier specifying groups of users that are associated in a group. This identifier is particularly important for virtual private optical networks (VPONs).
Priority	protection and restoration: priority, protection and represented by two attributes: service type and side protection
Service type	A class of service dealing with protection plan options; pertains to priority of connection robustness

associated with the create request and response messages. Some of these parameters are also coded in the modify, delete, status, and notification messages.

- Modification: This procedure is used to modify an existing light-path. There are restrictions on what can be modified, and include only GLSP-related parameters: (a) priority value, user-group ID, and maximum signaling delay.

- Deletion: This procedure reverses the creation procedure and tears down the lightpath. It is destructive, and the create procedure must be executed if a ligthpath is needed once again. It needs only the following parameters in its messages: (a) source and destination points, (b) GLSP ID, and (c) result code.
- Status: This procedure is used to provide status information between UNI and NNI nodes about the results of the other procedures. It reports on the success or failure of ongoing operations, such as a creation request.
- Notification: This service is available only at the UNI, and is used by the UNI-C to notify the NNI node about the status of a G.LSP at the UNI.

SUMMARY

The success of the ODSI and OIF work, and the associated IETF activities in relation to the UNI and NNI specifications, remain to be seen. Several vendor interoperability tests by the ODSI and OIF have been successfully completed, and by-and-large they have met with favorable reactions in the industry. I refer you to [ODSI01] and [OIF01] for more information of UNI and NNI operations.

14

ATM vs. IP in Optical Internets

This chapter discusses the issues surrounding the use of IP in optical networks. Since many existing 2G transport backbone networks run IP over ATM, the chapter examines the pros and cons of IP over optical vs. ATM over optical. (The chapter assumes you are familiar with ATM.)

Framing and encapsulation are important operations in an internet as well as in an optical network, and we analyze the pros and cons of several encapsulation and framing standards published by the ITU-T and the IETF.

IP OVER ATM OVER SONET

The prevalent approach today for moving IP traffic over a wide area network (WAN) is to use the services of ATM and SONET, as shown in Figure 14–1. This practice is called IP over ATM over SONET since IP is a layer 3 protocol, operating over ATM and SONET.

For these services to operate correctly, and for different vendors to be able to interwork their products together, many rules are needed. Here are some examples:

- Mapping of IP addresses into ATM virtual circuits
- Mapping of ATM cells into SONET payloads

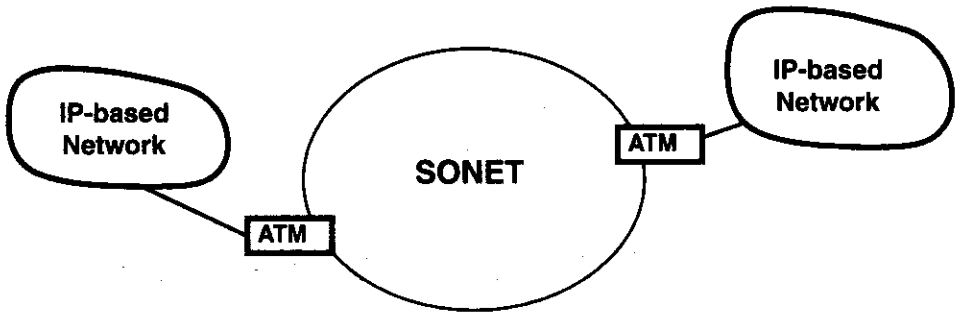


Figure 14–1 Running IP over ATM over SONET.

- Boundary alignments (octet alignments) of IP and ATM octets in the SONET payload envelope
- Correlation (if necessary) of ATM alarms to SONET OAM messages (of which there are many)
- Agreement on specifications for the IP to ATM encapsulation headers

THE OSI AND INTERNET LAYERED MODELS

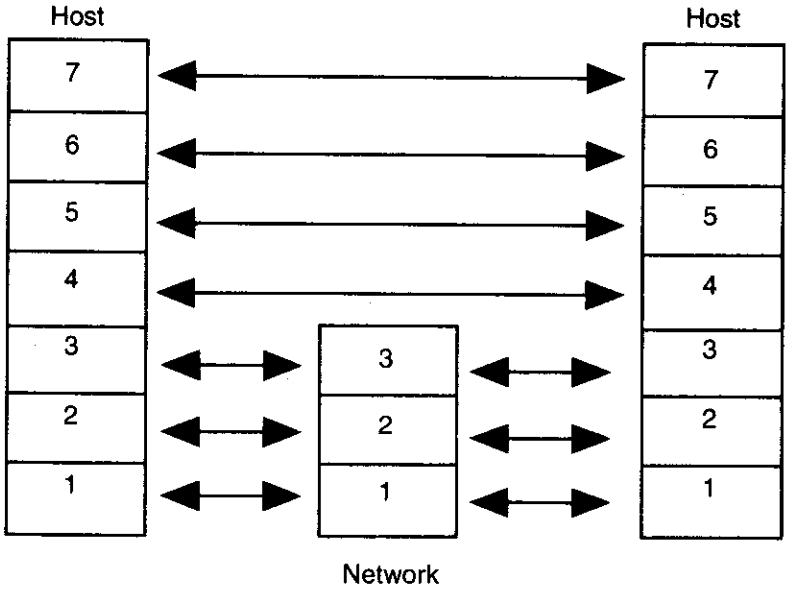
For the analysis of several subjects in this chapter, it will prove helpful to use the OSI and Internet layered reference models as a reference. Figure 14–2 shows the models.

The layers of the OSI reference model, the Internet model, and the layers of vendor's models, such as IBM's Systems Network Architecture (SNA), contain the communications functions at the lower three layers. From the OSI perspective, it is intended that the upper four layers reside in the host computers.

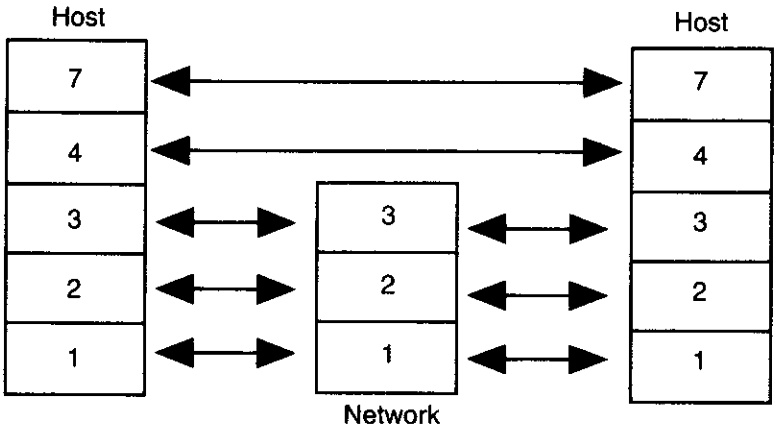
This does not mean that the lower three layers reside only in the network. In order to affect complete communications, the services in the lower three layers also exist at the host machine. End-to-end communications, however, occur between the hosts by invoking the upper four layers, and between the hosts and the network by invoking the lower three layers. This concept is shown in Figure 14–2(a) with the arrows drawn between the layers in the hosts and the network.

The "network" in the figure means the placement of the layers in perhaps hundreds of components (such as ATM switches and IP-based routers), although only one is shown.

Also, although the upper layers are not shown residing in the network nodes (such as routers), they do indeed exist in these components. But they are usually not invoked for the ongoing transport of user payload. They are executed to support operations within the network, such as setting up sessions between routers. In some situations, the upper



(a) The OSI Model

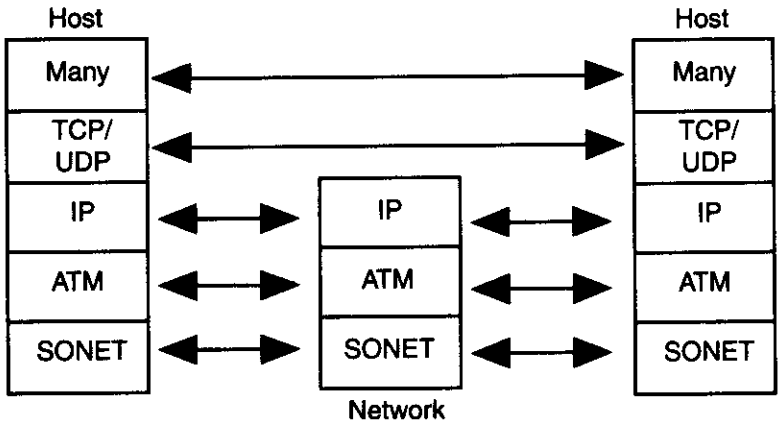


(b) The Internet Model

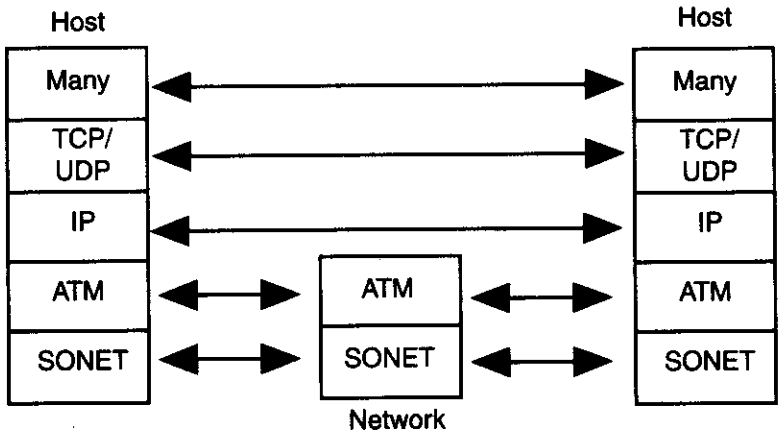
Figure 14–2 The OSI and Internet layered models.

layers are invoked between the network node and the host; one example is the establishment of TCP sockets between a host and a router in a mobile network for the support of traffic integrity operations, such as flow control, sequencing, and acknowledgments of traffic.

Figure 14–2(b) shows the Internet model. It does not include layer 5, the session layer. With rare exceptions, it does not include layer 6, the presentation layer, although the Simple Network Management Protocol (SNMP) uses a subset of layer 6. Therefore, layers 5 and 6 are not shown in Figure 14–2(b).



(a) The Internet Core Protocols (and SONET)



(b)With ATM as a Bearer Service

Figure 14–3 Typical protocol placements.

Placement of Core Protocols

Figure 14–3(a) shows the placements of the Internet core protocols in the Internet model. The two lower layers are operating with ATM and SONET. IP is operating at layer 3, with TCP/UDP at layer 4. At layer 7, a wide variety of protocols exists to support applications such as email, file transfer, network management, route discovery, Web browsing, etc.

Figure 14–3(b) shows a typical protocol stack in which IP is not used for IP address-based forwarding. In IP's place is ATM. This approach is considerably more efficient than the network protocol stack in Figure 14–3(a) because ATM replaces IP as the forwarding protocol. Virtual circuit switching using ATM is preferable to IP address forwarding because it is much faster.

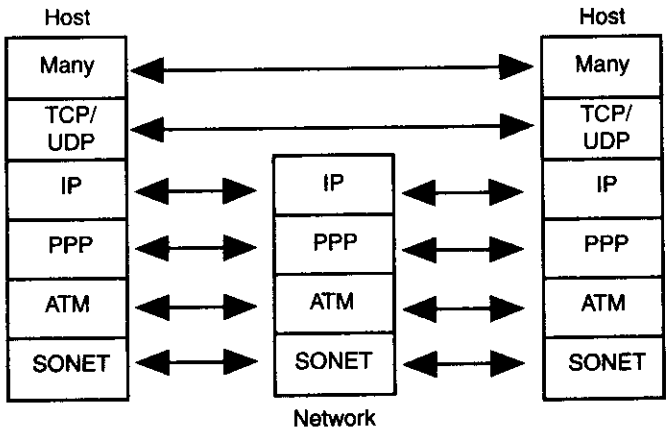
PPP and L2TP

The point-to-point protocol (PPP), shown in Figure 14–4(a), is widely used in the Internet and intranets. PPP is used to encapsulate network layer packets over a serial communications link. The protocol allows two machines on a point-to-point communications channel to negotiate the particular types of network layer protocols (such as IP) that are to be used during a session. It also allows the two machines to negotiate other types of operations, such as the use of compression and authentication procedures. After this negotiation occurs, PPP is used to carry the network layer protocol data units (PDUs) in the I field of an HDLC-type frame.

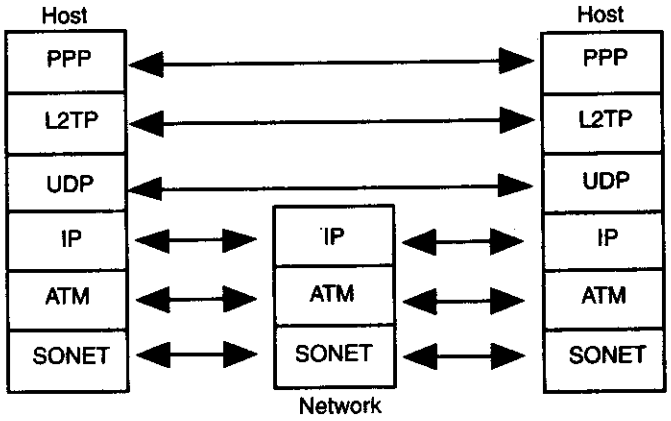
The layer 2 tunneling protocol (L2TP), shown in Figure 14–4(b), was introduced to allow the use of the PPP procedures between different networks and multiple communications links. With L2TP, PPP is extended as an encapsulation and negotiation protocol to allow the transport of PPP and user traffic between different networks and nodes.

One principal reason for the advent of L2TP is the need to dial in to a network access server (NAS) that may reside at a remote location. While this NAS may be accessed through the dial-up link, it may be that the NAS is located in another network. L2TP allows all the PPP operations to be used between machines in different networks. With the implementation of L2TP, an end user establishes a layer 2 connection to an access concentrator such as a modem bank, an ADSL bank, etc. Thereafter, the concentrator is responsible for creating a L2TP tunnel and sending the specific PPP packets to a network access server (NAS).

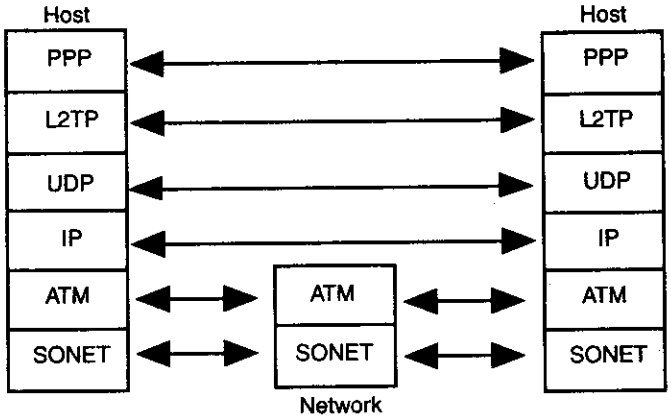
Figure 14–4(c) is similar to a previous layered stack and shows a typical protocol stack in the network if IP is not used for IP address-based forwarding. In IP's place is ATM. L2TP is used in this stack.



(a) Using PPP

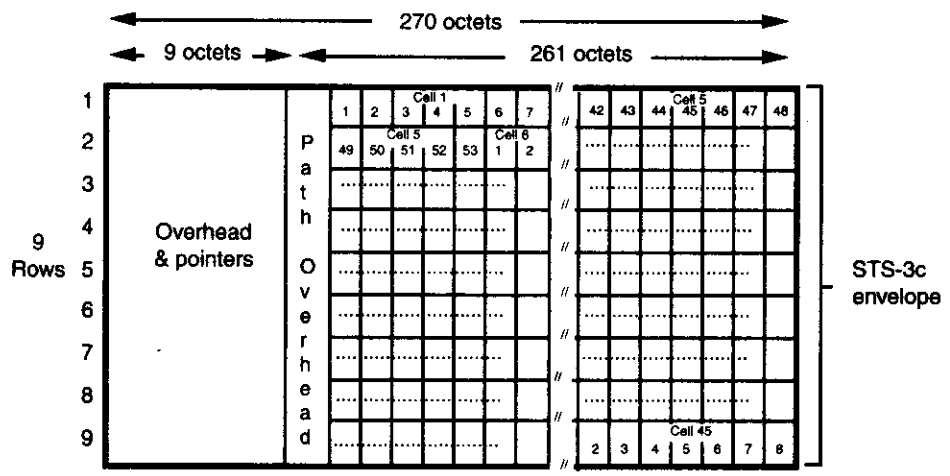


(b) Using PPP and L2TP



(c) With ATM as the Forwarding Protocol

Figure 14–4 Using PPP and L2TP.



Note: Example shows the first cell aligned exactly in the beginning of the payload area. It may be positioned anywhere in the payload.

Figure 14–5 ATM over SONET or SDH.

ATM IN THE SONET/SDH PAYLOAD ENVELOPE

Figure 14–5 shows how ATM cells are mapped into a SONET or SDH payload envelope. The payload pointers can be used to locate the beginning of the first cell. Additionally, cell delineation is achieved by the receivers locking onto the 5 bytes that satisfy the HEC operations. In this manner, the receiver knows where a cell is positioned in the envelope. The receiver also is able to detect an empty cell.

It is unlikely that cells would be positioned at the first byte of the payload. If they are, an STS-3c system can carry 44 cells, and bytes 1–8 of the 45th cell. The remainder of the 45th cell is placed in the next SONET frame. So, a cell can cross the tributary/container frame boundary.

PPP IN THE SONET PAYLOAD ENVELOPE

Request for comments (RFC) 1619 defines the rules for running the point-to-point protocol (PPP) over SONET. See Figure 14–6. Since SONET is a physical point-to-point circuit, PPP over SONET should be a straightforward operation. This section paraphrases RFC 1619 (which is quite terse).

There are ambiguities in this RFC, and some implementers have complained to me about some difficulty in using it as an authoritative guide. One complaint deals with the rules on octet alignment of the PPP payload in the SONET SPE. Anyway, listed below are the major rules as outlined in RFC 1619:

- PPP treats the SONET network as octet-oriented synchronous links.
- The PPP octet stream is mapped into the SONET synchronous payload envelope (SPE), with the PPP octet boundaries aligned with the SPE octet boundaries.
- Scrambling is not used.
- The path signal label (C2) is intended to indicate the contents of the SPE. The experimental value of 207 (cf hex) is used to indicate PPP.
- The multiframe indicator (H4) is currently unused and must be zero.
- The basic rate for PPP over SONET is that of STS-3c at 155.520 Mbit/s.
- The available information bandwidth is 149.760 Mbit/s, which is the STS-3c SPE with section, line, and path overhead removed. This operation is the same mapping used for ATM and FDDI.

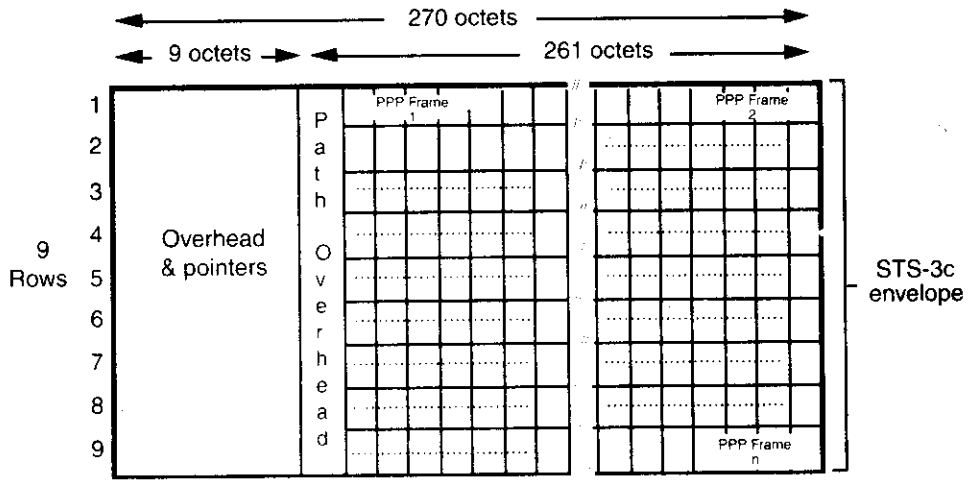


Figure 14–6 PPP over SONET.

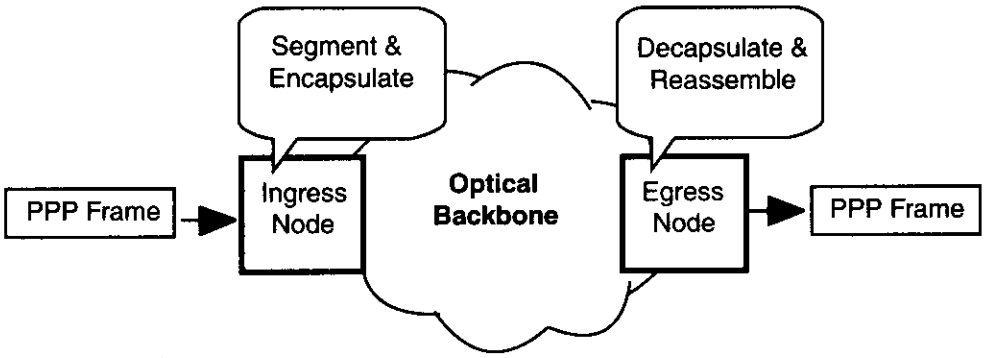


Figure 14–7 Operations at the Ingress and Egress nodes.

PREVALENT APPROACH IN TODAY'S INTERNETS

Before IP and PPP are encapsulated into SONET, they (typically) are first encapsulated into an ATM cell. Figure 14–7 shows that the IP/PPP traffic (the PPP frame) is accepted at the ATM switch, which is acting as the ingress node to the optical network. At this node, the PPP frame is segmented into ATM cells. The ATM switch also adds an encapsulation header; it is used to identify the type of traffic being carried in the cell, such as PPP, AppleTalk, SNA, etc. The ATM cell is then encapsulated into the optical payload, such as a SONET payload envelope.

At the egress node, the process is reversed. The SONET header, as well as the ATM header, is stripped away, leaving only the original PPP frame. The 48-byte segments are reassembled into the PPP frame. This frame is the same image as the frame presented to the ingress switch.

ENCAPSULATION/FRAMING RULES

As noted, before IP and PPP are encapsulated into SONET, they are first encapsulated into an ATM cell. Figure 14–8 shows the relationship of running IP over ATM, with emphasis on the CP-AAL5 and the ATM layers. ATM adaptation layer, type 5 (AAL5) performs its conventional segmentation and reassembly functions by delineating the traffic into 48-byte data units with the addition of an 8-byte trailer as part of the last data unit.

The error detection operation is provided by the AAL5 CRC-32 calculation over the PDU, and the padding field (PAD) is used to fill in the CPCS SDU to an even increment of 48 bytes.

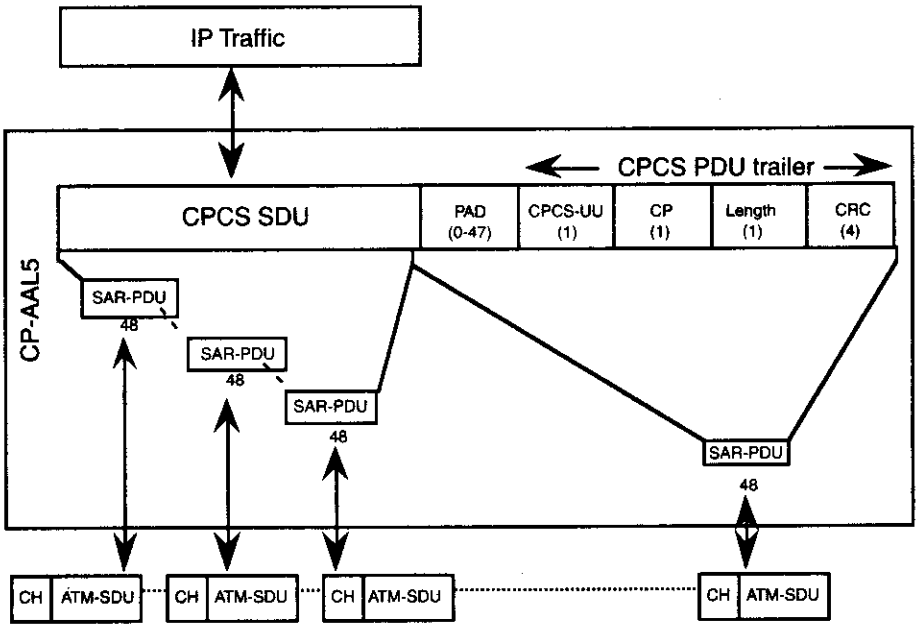


Figure 14–8 Classical IP over ATM.

ATM and Frame Relay Framing Formats

While the emphasis thus far at layer 2 has been on ATM, Frame Relay should not be excluded, because it has a big presence in the industry. The emphasis will remain on ATM, but we will include some discussions on Frame Relay as well.

Figure 14–9 shows the formatting and identification conventions for Frame Relay frames and AAL5 common part convergence sublayer (CPCS) PDUs with IP traffic. The Frame Relay frame and the AAL5 CPCS PDU use several industry standards for these operations. They are:

- *Control:* The control field, as established in HDLC standards. It is 1 byte in length.
- *NLPID:* The network level protocol ID, as established in the ISO/IEC TR 9577 standard. It is 1 byte in length.
- *OUI:* The organizationally unique ID, as established in RFCs 826, 1042, and cited in many other RFCs. It is 3 bytes in length.
- *PID:* The protocol ID, published as an Internet standard. It is also called Ethertype. It is 2 bytes in length.

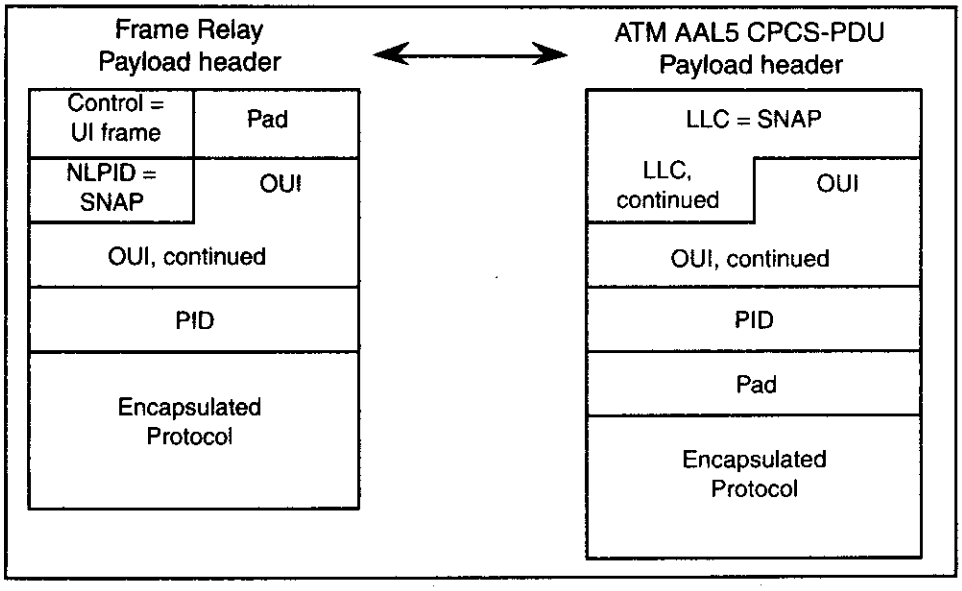


Figure 14–9 Formatting and identification conventions.

- *LLC:* The logical link protocol, as established in the IEEE 802.x standards. It is 3 bytes in length and includes three 1-byte fields: (a) source service access point (SAP), (b) destination SAP, and (c) the HDLC control field. The SAPs are usually set to 0xAA, and are not used (they defer to the subnetwork access protocol (SNAP)).
- *SNAP:* A header that includes OUI and PID.

The convention for the figures that follow is: one line entry in the figure represents two bytes. The exception is the last entry, which is payload and is variable in length.

If it appears to you that some of these fields perform redundant operations, your perception is correct. Due to the fragmented evolution of encapsulation headers, different groups have developed their own standards. In some situations, the result is that one encapsulation header identifies another encapsulation header. For example, in some systems the ISO NCPID identifies a SNAP header. Then the SNAP header identifies the OUI and PID fields. To be charitable, it is messy.

Encapsulation Field Values

As depicted in Figure 14–10, for the encapsulation of IP datagrams, the NLPID in the Frame Relay payload header of 0xCC (the reserved

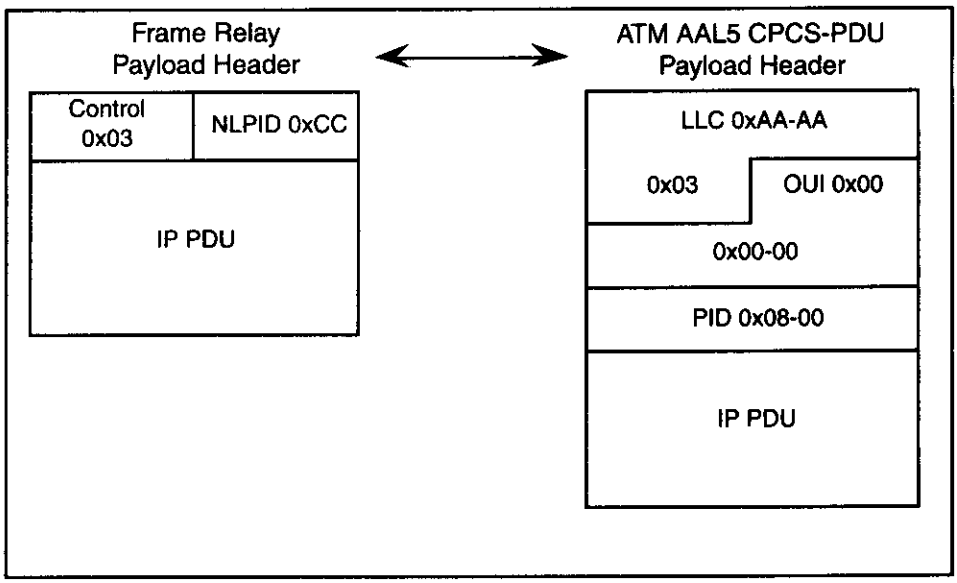


Figure 14–10 Frame relay/ATM payload header for IP PDUs.

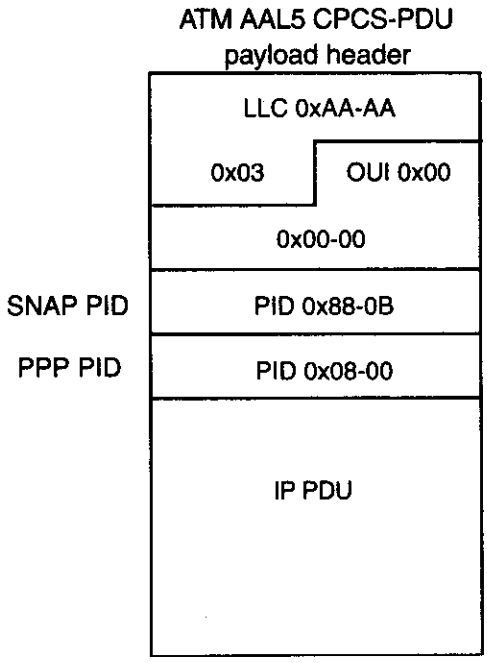
NLPID value for IP) performs the same functions as the PID value of 0x08–00 (the reserved PID value for IP). In the AAL5 CPCS-PDU payload header, the OUI is set to 0x00–00–00.

Encapsulation Options with SNAP

The encapsulation methods are an important part of the debate regarding ATM vs. IP over SONET. Therefore, it is a good idea to look at several encapsulation alternatives.

Since many systems run IP over PPP, a logical approach is to use the ATM SNAP header (the PID field) to identify that PPP follows IP. This ID is a registered Internet number of 0x88–0B, reserved for PPP. This value is also the registered Ethertype value for PPP. This idea is shown in Figure 14–11. (Note that one header identifies another header, and so on. Again, a bit messy.)

An important aspect of this scenario is the fact that PPP headers and trailers must be taken into account as part of the overall overhead. It can be confusing, for the following scenarios (among others) can exist:



Note: The other PPP header and trailer fields are not shown here. They reside in the area titled "PPP PID."

Figure 14–11 Using SNAP to identify PPP PID.

- IP over PPP over ATM over SONET
- IP over ATM over SONET
- IP over (just) PPP's PID over ATM over SONET

The third item can be a point of confusion. Let's clarify this entry. Some of the PPP overhead fields may not be used:

- *Beginning and Ending Flags:* Usage depends on implementations.
- *Address:* Not used, set to all 1s (but may be examined in some systems).
- *Control:* Not used, set to HDLC's unnumbered information (UI) (but may be examined in some systems).
- *FCS:* Usage depends on implementation.

You could state, "Why does it matter? If PPP is in both the IP and IP/ATM stacks, then it's a wash." Not quite; in the ATM stack, the spe-

cific implementation of PPP may push the payload into another ATM cell. If only a small part of this cell's 48-byte payload is needed, the remainder is filled with the padding bytes. This situation leads to a lot of overhead. On the other hand, if there is little or no padding needed, the situation is not a serious problem. We return to these issues later in the chapter.

THE PPP PACKET

Let's take a look at the PPP packet. The PPP packet uses the HDLC frame as stipulated in ISO 3309–1979 (and amended by ISO 3309–1984/PDAD1). Figure 14–12 shows this format. The flag sequence is the standard HDLC flag of 01111110 (0x7E); the address field is set to all 1s (Hex FF) which signifies an all stations address. PPP does not use individual station addresses because it is a point-to-point protocol. The control field is set to identify a HDLC unnumbered information (UI) command. Its value is 00000011 (0x03). The I (information) field carries the fields of the upper layer protocols, typically IP, UDP TCP, and a layer 7 protocol.

The protocol field is used to identify the traffic that is encapsulated into the I field of the frame, such as IP, OSPF, etc. The field values are assigned by the Internet, and the values beginning with a 0 identify the network protocol that resides in the I field. Values beginning with 8 identify a control protocol that is used to negotiate the protocols that will

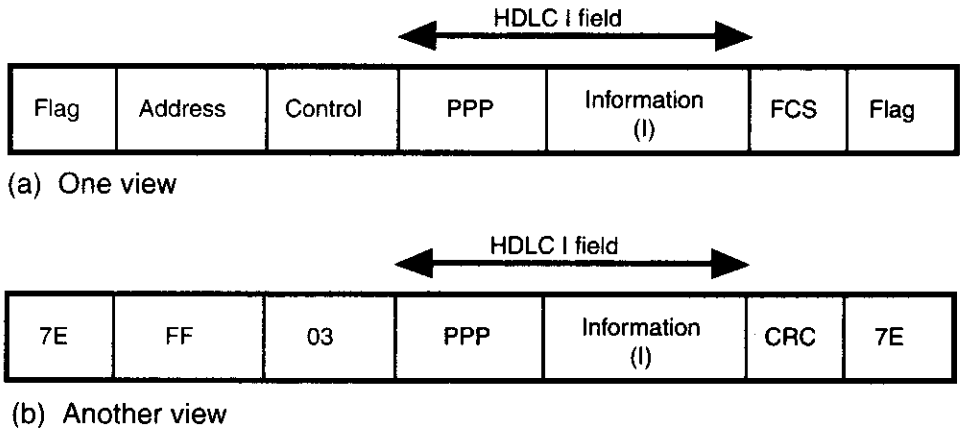


Figure 14–12 The PPP format.

actually be used. For example, IP is identified with the value of 0021, and the protocol used to negotiate IP operations during a PPP setup is 8021 (the IP control protocol).

THE ATM VS. IP DEBATE

Table 14–1 provides a comparison of IP and ATM. The Attributes column defines the capability of the attribute, and the next two columns explain how the two protocols support or do not support the attribute.

Table 14–1 Comparisons of IP and ATM

Attribute	IP	Cell Relay (ATM)
QOS support?	Very little (Note 1)	Extensive
Application support?	Asynchronous data (not designed for voice)	Asynchronous, synchronous voice, video, data
Connection mode?	Connectionless	Connection-oriented
Congestion management?	None (Note 2)	Extensive
Identifying traffic? (Note 3)	IP address	Virtual circuit ID: The VPI/VCI and an OSI address
Congestion notification?	None	The CN bits in the PTI field
Traffic tagging?	None (Note 4)	The cell loss priority (CLP) bit
PDU size?	Variable (a datagram)	Fixed at 48 bytes (a cell)
Sequence numbers	None	Cell header, no; for payload, depends on payload type
ACKs/NAKs/ Resends?	None	Only for signaling traffic (SVCs)
Protection switching	Not defined	Yes
Location?	In user machine (PC, etc.) and in routers and switches	Rarely in user machine, usually in routers and switches
Marketplace?	Quite prevalent	Prevalent

Note 1: The addition of MPLS and DiffServ/RSVP changes "Very little" to "Extensive."
Note 2: The addition of DiffServ and RSVP-TE changes "None" to "Extensive."
Note 3: For ATM, addresses are used initially for the virtual circuit provisioning. Thereafter, virtual circuit IDs are used. For IP, the DiffServ codepoint and/or the MPLS label can be used in place of the cumbersome IP address.
Note 4: The addition of DiffServ changes "none" to "extreme."

This table is important in regard to the IP vs. ATM debate. It shows that many capabilities that ATM provides are not available with IP. The conclusion drawn is that IP cannot be a direct replacement for ATM. It follows that removing ATM from the SONET stack translates into substantially reduced features and services to the user.

ATM opponents consider ATM too expensive. ATM is indeed "pricey." Yet the issue is not as simple as it may seem. ATM provides a very wide array of services, well beyond what IP provides (which is very little). If IP is going to provide ATM-like features (and most people agree that they are needed), several supporting protocols must be added to IP, such as MPLS and DiffServ.

If dynamic call processing is to be provided for telephone and video users, services like ATM's switched virtual call (SVC) must be added to the IP arsenal. This means protocols such as Megaco, MGCP, and SIP must become part of the IP network.

ATM is going to be a prevalent technology for quite some time. But it should also be emphasized that the combinations of IP/MPLS/DiffServ/RSVP-TE and several other supporting technologies can replace ATM completely, and provide all the functionality that ATM now provides.

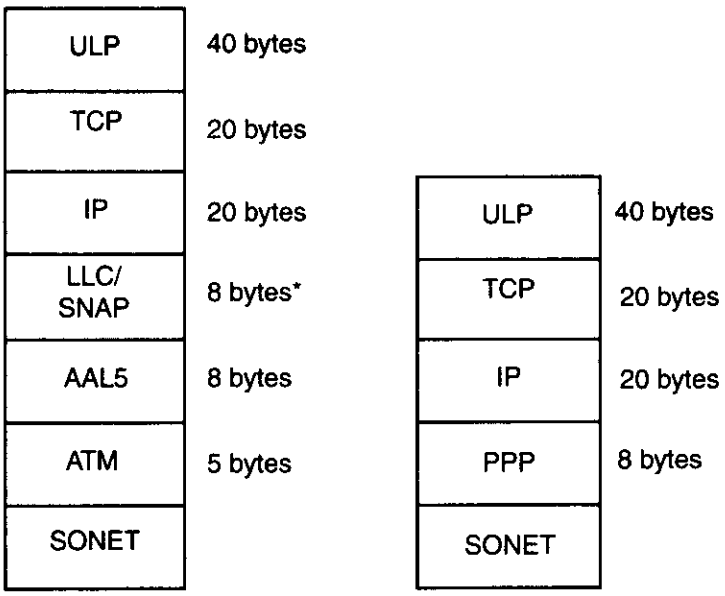
OVERHEAD OF IP AND ATM

As noted, one of the big issues today is the overhead of running IP over ATM versus the overhead of IP over an alternative, say, directly over SONET. Figure 14–13 provides some general information on the differences in the number of bytes needed for these two approaches.

The payload is assumed to be 40 bytes, to keep the comparisons consistent. Both protocol stacks have the same overhead at the upper layers since they are both supporting the same upper layer protocols, TCP, and IP.

The difference lies at the layers below IP. ATM uses the 3-byte LLC and the 5-byte SNAP (containing the 3-byte OUI and the 2-byte PID). The IP-only stack does not use these headers.

Please note the "note" in Figure 14–13. The ATM stack may not have to carry the full PPP headers and trailers across the network. It can extract the PPP PID field and map it into the SNAP PID field, then reconstruct the PPP headers and trailers at the egress to the network, if needed. This may be feasible since the flags, address fields, and control fields are present, but never change. Whether this is possible depends on the vendor's design of the ATM node.



* Note: If PPP is under IP, the SNAP PID can so indicate.

Figure 14–13 Comparisons of overhead.

Also, note that the total number of bytes passed to ATM is more than a 48-byte payload. Consequently, the traffic must be placed in more than one ATM cell; the details on this operation are shown next.

Is the ATM Overhead Tolerable?

The answer to the question "Is the ATM overhead tolerable?" will depend upon who supplies the answer. If the question is answered by a network manager on a private campus, the answer is likely to be yes. This manager is probably not paying hard dollars for bandwidth from a public WAN backbone operator. The attractive features of ATM may make the overhead less of an issue.[1]

[1]The same answer (yes) might come from a user of ATM who does not know ATM is being used. An example: some of the DSL modems being installed today run ATM between the user's site and the service provider's site. A more likely answer from this person would be, "I have no idea." I am one of those DSL users, and I do not like the fact that my DSL local loop is running ATM, because I know of the overhead it consumes. I also want to know what my service provider is doing with ATM in my DSL node.

The major problem occurs when a backbone customer is paying for bandwidth and is running the ATM stack. For example, ISPs pay for this overhead when they use a carrier's links. Their incentive is to eliminate ATM and run IP directly over SONET.

Another factor is the overhead of processing AAL, even though AAL5 should not be processed in the backbone network. Interfaces with OC-12 (STS-12c) now have ATM SAR chips. Interfaces are now available for OC-48 (STS-48c) for support of direct PPP/HDLC mappings.

THREE ENCAPSULATION METHODS

In this part of our analysis, we examine three methods of encapsulation:

1. Running IP and a full PPP frame over ATM
2. Running IP and only the PID field of PPP over ATM
3. Running IP and a full PPP frame directly over the physical layer

Method 1: Conventional Approach

For method 1, three ATM cells are needed to transport a 40-byte payload. Figure 14–14 shows why. First, 96 bytes are presented to AAL5. Previous discussions explains how these bytes are used (40 for ULP, 20 for TCP, 20 for IP, 8 for PPP, and 8 for LLC and SNAP (the OUI and PID)).

This unit has the AAL5 trailer appended to it and is divided into 48-byte segments. The padding field is used to round out the traffic to an even increment of 48 bytes. With the 96 bytes and the 8 bytes contributed by AAL5, the unit is 104 bytes. Thus the padding field is 40 bytes (104 + 40 = 144 bytes / 48 bytes per cell = 3 ATM service data units (SDUs). The third SAR-PDU contains only the 8-byte AAL5 trailer and 40 bytes of padding. Next, the three 5-byte cell headers are appended to the service data units to yield a total of 144 + 15 = 159 bytes.

Recall that the original payload was 40 bytes of ULP, plus the 20 bytes each of the TCP and IP headers, the 8 bytes for PPP and the 8 bytes for the LLC and SNAP fields. Thus, the ratio of payload to overhead is 96:159.

Well, maybe; it depends on how one defines overhead. The LLC and SNAP (OUI and PID) should be considered overhead when counting

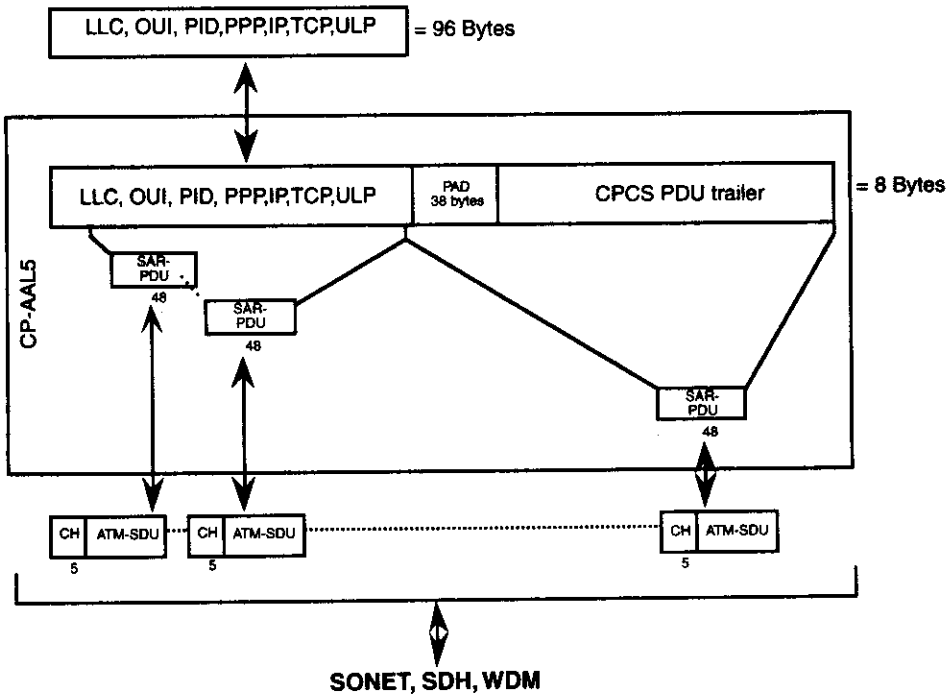


Figure 14–14 Operations to support Method 1.

bytes, since the ATM bearer service requires their use. If ATM is not used, LLC and SNAP are not needed. One could argue that the ratio of payload to overhead is 88:174 (minus the 8 bytes of LLC and SNAP). I think it fairer to use this latter ratio, especially when comparing ATM to IP. Consequently, in this example the use of ATM carries nearly an 80% overhead penalty.

Method 2: Lightweight PPP

For method 2, the PPP flags, address field, control field, and CRC field are stripped away before the payload is presented to CP-AAL5. Thus, 6 bytes are removed. The value used to identify PPP is carried in the ATM PID field (0x88–0B). Recall that this method requires the egress node to reconstruct these fields for presentation to the receiving application (if it needs them). These fields do not change, and the

ATM/SONET payload alignment can be performed with the SONET pointers or the ATM HEC field.

Let's see what it gives us; see Figure 14–15. The submitted unit to CP-AAL5 is now 90 bytes (because the two 1-byte flags, the 1-byte address field, the 1-byte control field, and the 2-byte CRC field are not present). The 8-byte AAL5 trailer is added to yield a unit of 98 bytes.

Three ATM cells are still needed: 98 / 48 = 2, with a remainder of 2 bytes. Therefore, the third cell contains 2 bytes of data, 38 bytes of padding, and the 8-byte AAL5 trailer.

So, deleting the PPP protocol control information did not reduce the number of ATM cells needed to support this IP packet.

However, if the system uses any one or a combination of three standardized options, the payload can be decreased substantially before it is presented to CP-AAL5, leading to the use of fewer than three ATM cells.

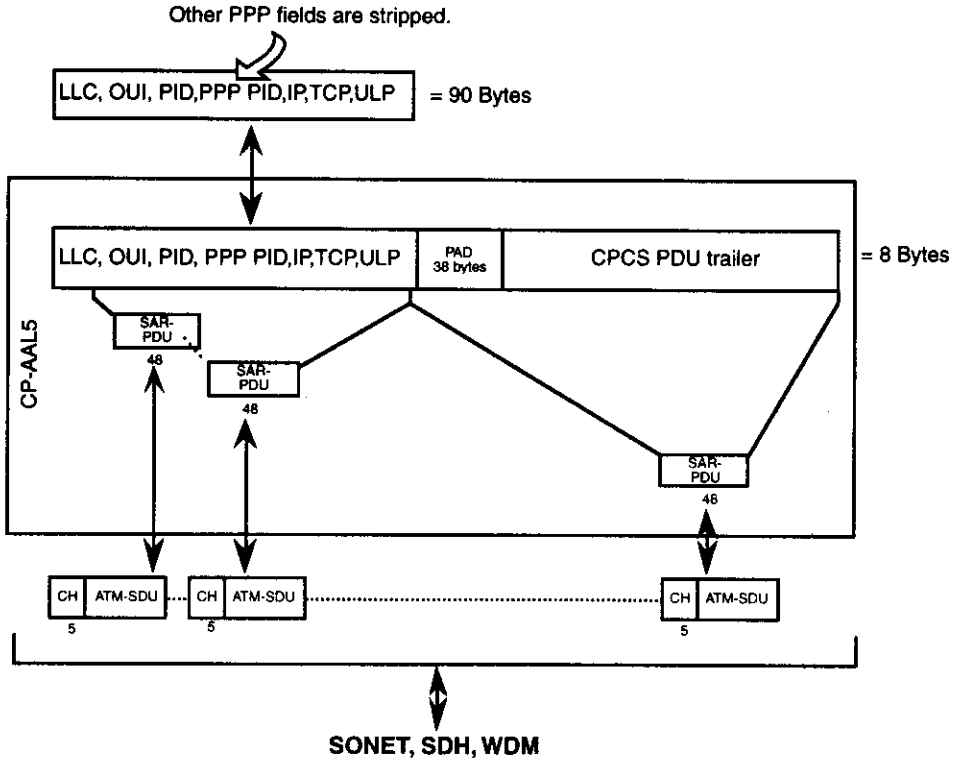


Figure 14–15 Operations to support Method 2.

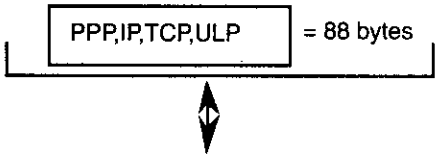


Figure 14–16 Operations to support Method 3.

These options are: (a) PPP PID field compression, (b) IP header compression, (c) TCP header compression.

Indeed, all methods are improved by the use of these techniques, not just method 2. Check your RFCs for information on these important compression operations.

Method 3: Eliminating ATM

For method 3, as illustrated in Figure 14–16, ATM is not used. Instead, PPP, IP, and the upper layer protocols are run directly over the physical layer. Obviously, this approach changes the payload-to-overhead ratio significantly. The overhead-associated LLC, SNAP, AAL5, and ATM are not needed.

For our analysis of these three methods, it might be concluded that there is no choice but to eliminate ATM. If the reduction of overhead is the only consideration, that is a valid conclusion. However, it was noted that ATM has many features, such as QOS traffic policing operations, and extensive diagnostic capabilities. If ATM is removed, so are its many features.

SUMMARY

The debate regarding IP vs. ATM has been going on for several years. In its simplest from, it revolves around the overhead of ATM and its fixed-length size. Whether or not these attributes of ATM are a handicap to its use is dependent on those that use ATM.

It is a good idea to check network provider tariffs and determine if any possible (and probable) ATM overhead is being charged back to you. I expect this discovery will affect your opinions about the ATM vs. IP debate.

But, regardless of this debate, IP is here to stay. ATM is here to stay as well, at least for a long time. Eventually, I think ATM will be replaced by MPLS and supporting protocols, such as DiffServ, CR-LDP, and RSVP-TE.