Network Working Group
Internet Draft
Expiration Date: May 2002

Andre Fredette (PhotonEx Corp.) (Editor)
Jonathan Lang (Calient Networks) (Editor)

Osama Aboul-Magd (Nortel Networks)
S. Brorson (Axiowave Networks)
S. Dharanikota (Nayna Networks, Inc)
John Drake (Calient Networks)
David Drysdale (Data Connection)
W. L. Edwards (iLambda Networks)
Adrian Farrel (Movaz Networks)
R. Goyal (Axiowave Networks)
Hirokazu Ishimatsu (Japan Telecom)
Monika Jaeger (T-systems)
R. Krishnan (Axiowave Networks)
Raghu Mannam (Hitachi Telecom)
Eric Mannie (Ebone (GTS))
Dimitri Papadimitriou (Alcatel)
Vasant Sahay (Nortel Networks)
Jagan Shantigram (PhotonEx Corp.)
Ed Snyder (PhotonEx Corp.)
George Swallow (Cisco Systems)
G. Tumuluri (Calient Networks)
Y. Xue (UUNET/WorldCom)
Lucy Yong (Williams Communications)
J. Yu (Zaffire, Inc)

November 2001

Link Management Protocol (LMP) for DWDM Optical Line Systems

draft-fredette-lmp-wdm-03.txt

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026 [RFC2026].

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet- Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at

http://www.ietf.org/shadow.html.

ABSTRACT

A suite of protocols is being developed in the IETF to allow networks consisting of photonic switches (PXCs), optical crossconnects (OXCs), routers, switches, DWDM optical line systems (OLSs), and optical add-drop multiplexors (OADMs) to use an MPLS-based control plane to dynamically provision resources and to provide network survivability using protection and restoration techniques. As part of this protocol suite, the Link Management Protocol (LMP) [LMP] is defined to "maintain control channel connectivity, verify component link connectivity, and isolate link, fiber, or channel failures within the network." In it's present form, [LMP] focuses on peer communications (eg. OXC-to-OXC). In this document we propose extensions to LMP for use with OLSs. These extensions are intended to satisfy the "Optical Link Interface Requirements" described in [OLI].

CONTENTS

SUMMARY FOR SUB-IP RELATED INTERNET DRAFTS
(Section Requested by Bert and Scott)

SUMMARY

This work is motivated by two main issues. The first is the need to enhance the fault detection and recovery support for photonic switches (PXCs), and the second is to enhance the discovery of link characteristics for optical networks in general.

GMPLS is being developed to allow networks consisting of photonic switches (PXCs), optical crossconnects (OXCs), routers, switches and optical line systems (OLS) (or DWDM systems) to use an MPLS-based control plane to dynamically provision resources and to provide network survivability using protection and restoration techniques. As part of this protocol suite, the Link Management Protocol (LMP) [LMP] is defined to "maintain control channel connectivity, verify component link connectivity, and isolate link, fiber, or channel failures within the network." In it's present form, [LMP] focuses on peer communications (e.g., OXC-to-OXC). In this document we propose extensions to LMP for use with optical line systems. These extensions allow the OLS to inform attached devices, such as routers or PXCs, of (1) link properties needed for routing/signalling and (2) link failures that can be used to drive failure recovery protocols.

RELATED DOCUMENTS

http://www.ietf.org/internet-drafts/draft-ietf-ccamp-lmp-02.txt
http://www.ietf.org/internet-drafts/draft-many-oli-reqts-00.txt

WHERE DOES IT FIT IN THE PICTURE OF THE SUB-IP WORK

lmp-wdm fits in the Control part of the sub-ip work.

WHY IS IT TARGETED AT THIS WG

lmp-wdm enhances the ability of circuit switches and routers using MPLS-based control protocols to dynamically discover link properties and to learn about link status. The link properties can be useful during signalling of paths, and the link status information is essential for fault detection and recovery. Furthermore, lmp-wdm is independent of any signalling protocol, so it can be used by both distributed control system, such as GMPLS, and centralized management systems.

Therefore, lmp-wdm supports the following CCAMP objectives:

. Define signalling protocols and measurement protocols such that they support multiple physical path and tunnel technologies (e.g., O-O and O-E-O optical switches, ATM and Frame Relay switches, MPLS, GRE) using input from technology-specific working groups such as MPLS, IPO, etc.

- Define signalling and measurement protocols that are independent of each other. This allows applications other than the signalling protocol to use the measurement protocol; it also allows the signalling protocol to use knowledge obtained by means other than the measurement protocol.

- Abstract link and path properties needed for link and path protection. Define signalling mechanisms for path protection, diverse routing and fast path restoration. Ensure that multi-layer path protection and restoration functions are achievable using the defined signalling and measurement protocols, either separately or in combination.

- Define how the properties of network resources gathered by the measurement protocol can be distributed in existing routing protocols, such as OSPF and IS-IS.

JUSTIFICATION

draft-fredette-lmp-wdm-03.txt (lmp-wdm) is a protocol proposal intended to satisfy the optical link interface (OLI) requirements (draft-many-oli-reqts-00.txt). There has been a great deal of interest in this work by both network operators and vendors, and the requirements document has achieved consensus in the CCAMP working group.

1. Introduction

Future networks will consist of photonic switches (PXCs), optical crossconnects (OXCs), routers, switches, DWDM optical line systems (OLSs), and optical add-drop multiplexors (OADMs) that use the GMPLS control plane to dynamically provision resources and to provide network survivability using protection and restoration techniques. A pair of nodes (e.g., a PXC and an OLS) may be connected by thousands of fibers. Furthermore, multiple fibers and/or multiple wavelengths may be combined into a single bundled link. [LMP] Defines the Link Management Protocol (LMP) to "maintain control channel connectivity, verify component link connectivity, and isolate link, fiber, or channel failures within the network." In it's present form, [LMP] focuses on peer communications (eg. OXC-to-OXC) as illustrated in Figure 1. In this document, extensions to LMP for use with OLSs are proposed. These extensions are intended to satisfy the "Optical Link Interface Requirements" described in [OLI]. It is assumed that the reader is familiar with LMP as defined in [LMP].

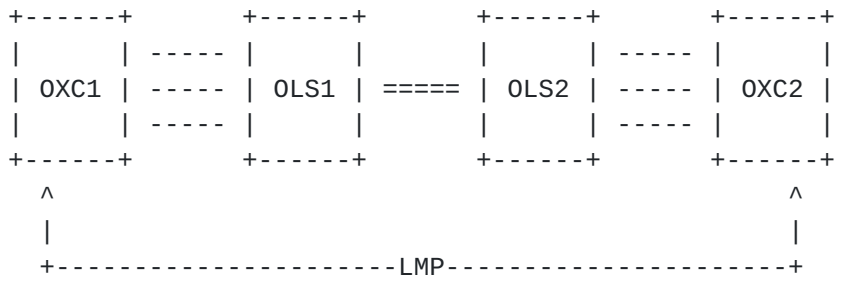


Figure 1: Base LMP Model

A great deal of information about a link between two OXCs is known by the OLS. Exposing this information to the control plane via LMP can improve network usability by further reducing required manual configuration and also by greatly enhancing fault detection and recovery. Fault detection is particularly an issue when the network is using all-optical photonic switches (PXC). Once a connection is established, PXCs have only limited visibility into the health of the connection. Even though the PXC is all-optical, long-haul OLSs typically terminate channels electrically and regenerate them optically, which presents an opportunity to monitor the health of a channel between PXCs. LMP-WDM can then be used by the OLS to provide this information to the PXC using LMP-WDM.

In addition to the link information known to the OLS that is exchanged through LMP-WDM, some information known to the OXC may also be exchanged with the OLS through LMP-WDM. This information is

useful for alarm management and link monitoring (i.e., trace monitoring). Alarm management is important because the administrative state of a connection, known to the OXC (e.g., this information may be learned through the Admin Status object of GMPLS signaling [GMPLS]), can be used to suppress spurious alarms. For example, the OXC may know that a connection is ôupö, ôdownö, in a

ôtestingö mode, or being deleted (ôdeletion-in-progressö). The OXC can use this information to inhibit alarm reporting from the OLS when a connection is ôdownö, ôtestingö, or being deleted.

The model for extending LMP to OLSs is shown in Figure 2.

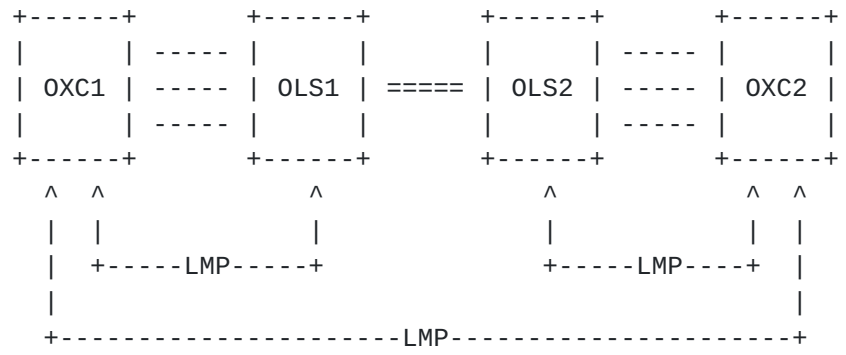


Figure 2: Extended LMP Model

In this model, an OXC may have multiple LMP sessions corresponding to multiple peering relationships. At each level, LMP provides link management functionality (i.e., control channel management, physical connectivity verification, link property correlation) for that peering relationship. For example, the OXC-OXC LMP session in Figure 2 can be used to build traffic-engineering (TE) links for GMPLS signaling and routing, and are managed as described in [LMP]. At the transport level, the OXC-OLS LMP session (also shown in Figure 2) is used to augment knowledge about the links between the OXCs. The management of these LMP sessions is discussed in this draft. It is important to note that an OXC may peer with one or more OLSs and an OLS may peer with one or more OXCs.

Although there are many similarities between an OXC-OXC LMP session and an OXC-OLS LMP session, particularly for control management and link verification, there are some differences as well. These differences can primarily be attributed to the nature of an OXC-OLS link, and the purpose of OXC-OLS LMP sessions. As previously mentioned, the OXC-OXC links can be used to provide the basis for GMPLS signaling and routing at the optical layer. The information exchanged over LMP-WDM sessions is used to augment knowledge about the links between OXCs.

In order for the information exchanged over the OXC-OLS LMP sessions to be used by the OXC-OXC session, the information must be coordinated by the OXC. However, the two LMP sessions are run independently and MUST be maintained separately. One critical requirement when running an OXC-OLS LMP session is the ability of

the OLS to make a data link transparent when not doing the verification procedure. This is because the same data link may be verified between OXC-OLS and between OXC-OXC. The BeginVerify procedure of [LMP] is used to coordinate the Test procedure (and hence the transparency/opaqueness of the data links).

To maintain independence between the sessions, it MUST be possible for the LMP sessions to come up in any order. In particular, it MUST be possible for an OXC-OXC LMP session to come up without an OXC-OLS LMP session being brought up, and vice-versa.

This draft focuses on extensions required for use with opaque transmission systems. Work is ongoing in the area of fully transparent wavelength routing; however, it is premature to identify the necessary characteristics to exchange. That said, the protocol described in this document provides the necessary framework in which to advertise additional information as it is deemed appropriate.

Additional details about the extensions required for LMP are outlined in the next section.

2. LMP Extensions for Optical Line Systems

As currently defined, LMP consists of four types of functions:

1. Control Channel Management
2. Link Verification
3. Link Summarization
4. Fault Management

All four functions are supported in LMP-WDM. Additionally, a trace monitoring function is added. (Note: Other monitoring types will be considered in a future release.)

In this document we follow the convention of [LMP] and use the term "data link" to refer to either "component links" or "ports".

It is very important to understand the subtle distinctions between the different types of links being considered in the extended LMP-WDM. For example, in Figure 2 when OXC1 and OXC2 complete the verify process, the links being verified are the end-to-end links between the OXC's. It is the TE link composed of these "data links" that are advertised in the routing protocols and used for the purposes of connection setup. The verify procedure between OXC1 and OLS1, on the other hand verifies the shorter link between these two nodes. However, each of these shorter links is a segment of one of the larger end-to-end links. The verify serves two functions: to verify connectivity and exchange handles by which each data link is referred. Furthermore, it is up to the OXC to correlate the handles between the various LMP sessions.

Once a control channel has been established and the OXC-OLS verification procedure has been completed successfully, the OXC and OLS may exchange information regarding link configuration (i.e., using the LinkSummary exchange). An OXC may also receive

notification regarding the operational status from an OLS (i.e., using the ChannelStatus exchange).

In subsequent sections, specific additions are proposed to extend LMP to work with OLSs.

2.1. Control Channel Management

As in [LMP], we do not specify the exact implementation of the control channel; it could be, for example, a separate wavelength or fiber, an Ethernet link, an IP tunnel through a separate management network, or the overhead bytes of a data link.

The control channel management for OXC-OLS links is the same as for OXC-OXC links, as described in [LMP]. The ôLMP-WDM Supportö flag in the LMP Common Header is used to indicate support for the objects defined in this draft. This informs the receiving node that the LMP-WDM extensions will be used for the session. If the LMP-WDM extensions are not supported by the node, it MUST reply to the Config Message with a ConfigNack Message.

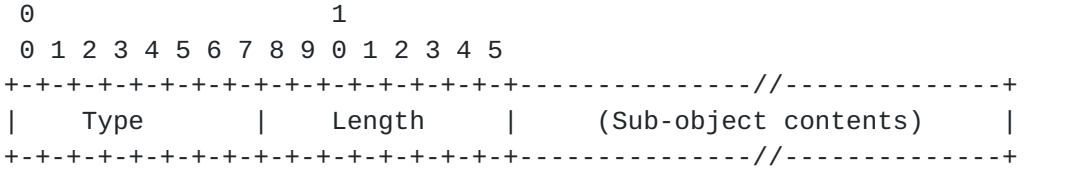
2.2. Link Verification

The Test procedure used with OLSs is the same as described in [LMP]. The VerifyTransportMechanism (included in the BeginVerify and BeginVerifyAck messages) is used to allow nodes to negotiate a link verification method and is essential for transmission systems which have access to overhead bytes rather than the payload. The VerifyId (provided by the remote node in the BeginVerifyAck message, and used in all subsequent Test messages) is used to differentiate Test messages from different LMP sessions.

2.3. Link Summarization

As in [LMP], the LinkSummary message is used to synchronize the Interface Ids and correlate the properties of the TE link. (Note that the term æTE LinkÆ originated from routing/signaling applications of LMP, whereas this concept doesnÆt necessarily apply to an OLS. However, the term is used in this draft to remain consistent with LMP terminology.) Additional Data Link sub-objects are defined in this draft to extend the LinkSummary message to include additional link characteristics. These sub-objects are described in the following subsections. The link characteristics, in general, are those characteristics needed by the control plane for constraint-based routing in the selection of a path for a particular connection.

The format of the Data Link Sub-Objects follows the format described in [LMP] and is shown below for readability:



Type: 8 bits

The Type indicates the type of contents of the subobject.

Length: 8 bits

The Length field contains the total length of the sub-object in bytes, including the Type and Length fields. The Length MUST be at least 4, and MUST be a multiple of 4.

The following Link Characteristics are advertised on a per data link basis.

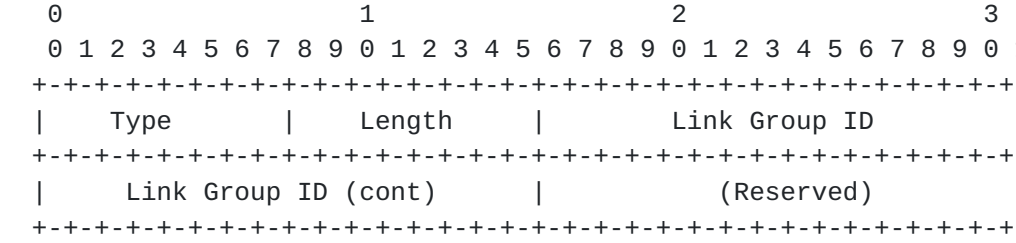
2.3.1. Link Group ID

The main purpose of the Link Group ID is to reduce control traffic during failures that affect many data links. A local ID may be assigned to a group of data links. This ID can be used to reduce the control traffic in the case of a failure by enabling the systems to send a single message for a group instead of individual messages for each member of the group. A link may be a member of multiple groups. This is achieved by presenting multiple Link Group ID Objects in the LinkSummary message.

The Link Group ID feature allows Link Groups to be assigned based upon the types of fault correlation and aggregation supported by a given OLS. From a practical perspective, the Link Group ID is used to map (or group) data links into "failable entities" known only to the OLS. If one of those failable entities fails, all associated data links are failed and the OXC may be notified with a single message.

For example, an OLS could create a Link Group for each laser in the OLS. This group could be associated with data links during discovery/initialization time. Multiple data links could be associated with a single group (depending on the kind of multiplexing supported in the system). If a laser fails, the OLS can report a failure for the group. The OXC that receives the group failure message can determine the associated link or links. Another group could be assigned for a fiber to report all data links down that are associated with that fiber if LOS is detected at the fiber level. Depending on the physical OLS implementation, it may make sense to allocate other groups, such as all data links on a particular circuit card. With this method, the OXC only needs to know about the externally visible data links. The OLS can associate the data links with logical groups and the OXC doesn't need to know anything about the physical OLS implementation or how data links are multiplexed electrically or optically within the system.

The format of the Link Group ID sub-object (Type=TBD, Length=8) is as follows:



Link Group ID: 32 bits

Link Group ID 0xFFFFFFFF is reserved and indicates all data links in a TE link. All data links are members of Link Group 0xFFFFFFFF by default.

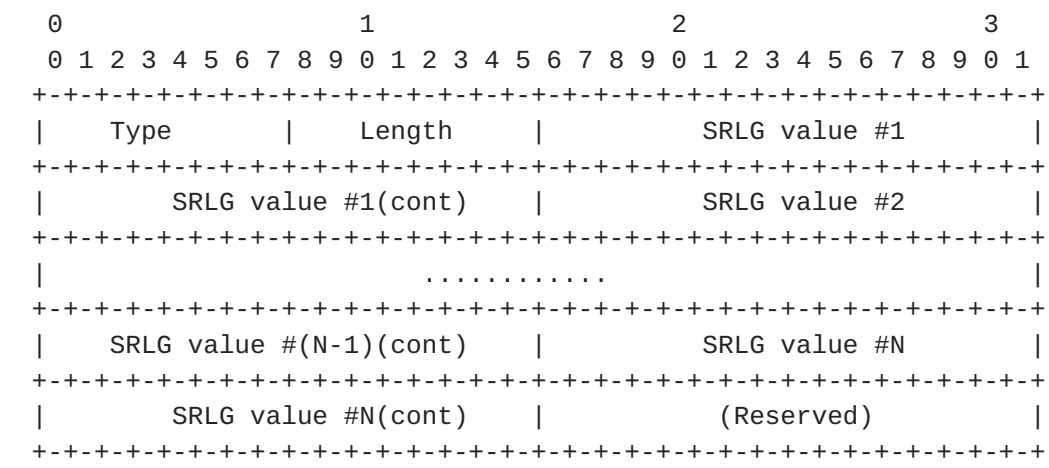
Reserved: 16 bits

Must be set to zero on transmit and ignored on receive.

2.3.2. Shared Risk Link Group Identifier (SRLG):

SRLGs of which the data link is a member. This information is manually configured on an OLS by the user and may be used for diverse path computation.

The format of the SRLG sub-object (Type=TBD) is as follows:



Length: 8 bits

The length is (N+1)*4, where N is the number of SRLG values.

Shared Risk Link Group Value: 32 bits

List as many SRLGs as apply.

Reserved: 16 bits

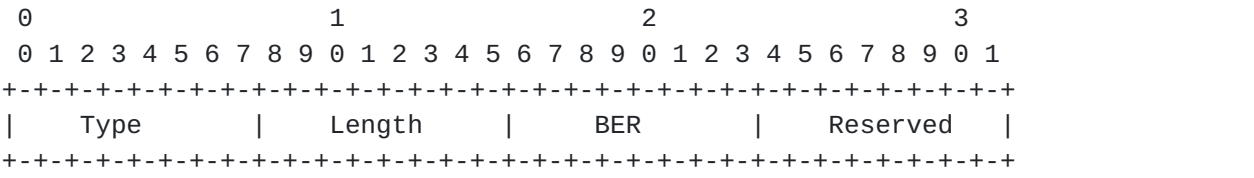
Must be set to zero on transmit and ignored on receive.

2.3.3. Bit Error Rate (BER) Estimate

This Object provides an estimate of the BER for the data link.

The bit error rate (BER) is the proportion of bits that have errors relative to the total number of bits received in a transmission, usually expressed as ten to a negative power. For example, a transmission might have a BER of "10 to the minus 13", meaning that, out of every 10,000,000,000,000 bits transmitted, one bit may be in error. The BER is an indication of overall signal quality.

The format of the BER Estimate subobject (Type=TBD; Length=4) is as follows:



BER: 8 bits

The exponent from the BER representation described above. For example, if the BER is 10 to the minus X, the BER field is set to X.

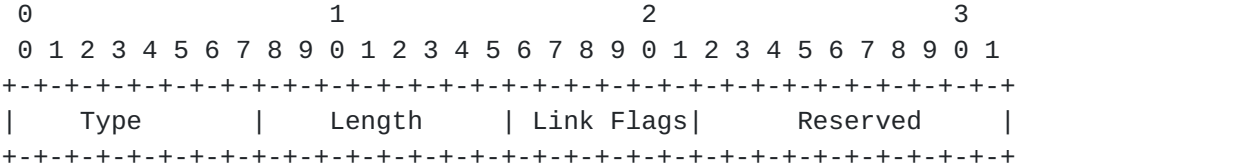
Reserved: 8 bits

Must be set to zero on transmit and ignored on receive.

2.3.4. Optical Protection

Whether the OLS protects the link internally. This information can be used as a measure of quality of the link. It may be advertised by routing and used by signaling as a selection criterion as described in [GMPLS].

The format of the Optical Protection subobject (Type=TBD; Length=4) is as follows:



Link Flags: 6 bits

Encoding for Link Flags can be found in [GMPLS].

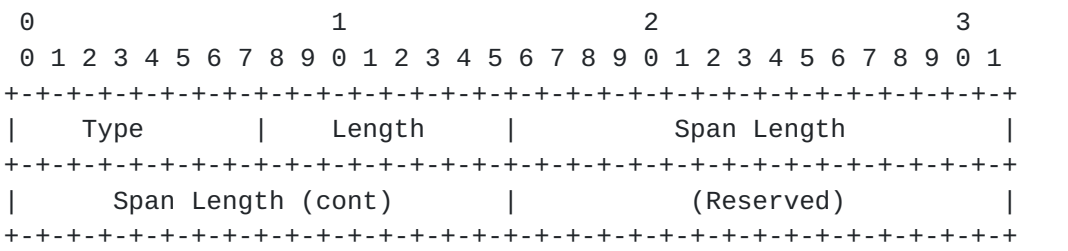
Reserved: 10 bits

Must be set to zero on transmit and ignored on receive.

2.3.5. Total Span Length:

The total distance of fiber in OLS. May be used as a routing metric or to estimate delay.

The format of the Span Length sub-object (Type=TBD, Length=8) is as follows:



Span Length: 32 bits

Total Length of the WDM span in meters expressed as an unsigned integer.

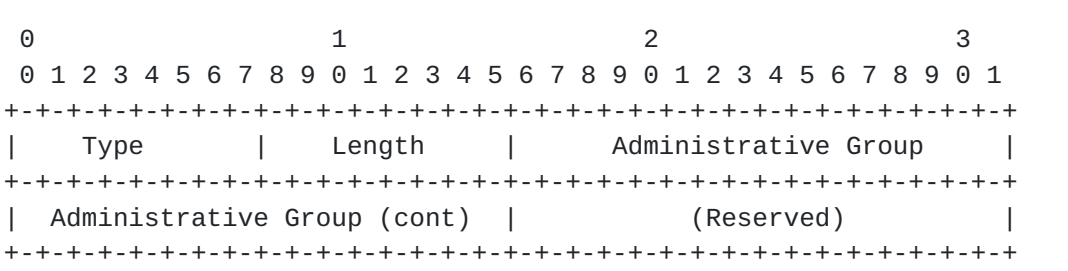
Reserved: 16 bits

Must be set to zero on transmit and ignored on receive.

2.3.6. Administrative Group (Color)

The administrative group (or Color) to which the data link belongs.

The format of the Administrative Group sub-object (Type=TBD, Length=8) is as follows:



Administrative Group: 32 bits

A 32 bit value.

Reserved: 16 bits

Must be set to zero on transmit and ignored on receive.

2.4. Fault Management

Fault management consists of three major functions:

1. Fault Detection
2. Fault Localization
3. Fault Notification

The actual Fault Detection mechanisms are the responsibility of the individual nodes and are not specified as part of this protocol. Fault detection mechanisms may include such things as bit error rate (BER) exceeding a threshold, loss of signal (LOS) and SONET/SDH-level errors. It is the responsibility of the OLS to translate these failures into OK, SF, or SD as described in LMP.

Running LMP-WDM on the OLS allows the OLS to notify an attached OXC or router when it detects a fault. The OXCs and routers continue to execute the fault localization procedure as currently specified in [LMP]. The main enhancement when using LMP-WDM is that the OLS may initiate the process (both downstream and upstream). It is important to note that the OLS does not participate in end-to-end fault localization as described in [LMP].

The OLS may also execute its own fault localization process that may allow it to determine the location of the fault much more specifically than the OXCs can. For example, the OLS may be able to pinpoint the fault to a particular amplifier along a set of fibers that can span 1000's of kilometers.

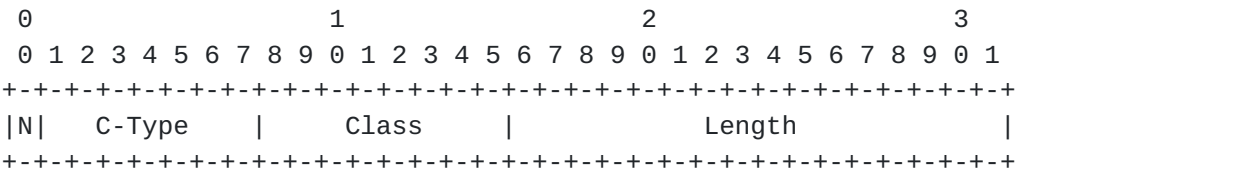
To report data link failures and recovery conditions, LMP-WDM uses the ChannelStatus, ChannelStatusAck, ChannelStatusRequest, and ChannelStatusResponse Messages defined in [LMP].

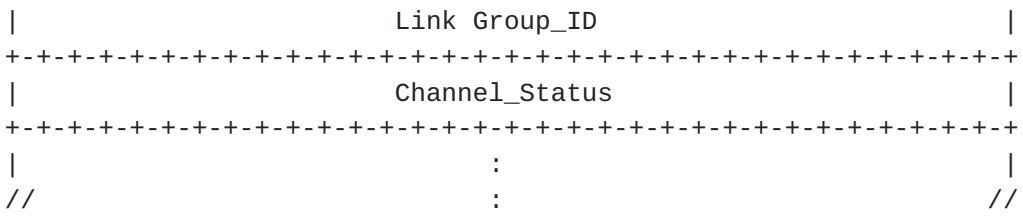
Each data link is identified by an Interface_ID. In addition, LMP-WDM specifies a Link Group_ID that may be assigned to a group of data links (see Section 2.3.1). The Link Group ID may be used to reduce the control traffic by providing channel status information for a group of data links. A new LINK GROUP_CHANNEL STATUS object is defined below for this purpose. This object may be used in place of the CHANNEL_STATUS objects described in [LMP] in the ChannelStatus message.

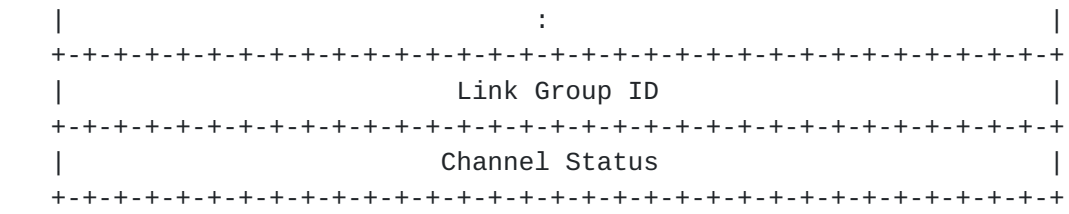
2.4.1. LINK GROUP CHANNEL_STATUS Object

The LINK GROUP_STATUS object is used to indicate the status of the data links belonging to a particular Link Group. The correlation of data links to Group ID is made with the Link Group ID subobject of the DATA_LINK Object.

The format of the LINK GROUP_CHANNEL STATUS object is as follows (Class = 18, C-Type to be assigned by IANA):




Link Group_ID
Channel_Status



Link Group_ID: 32 bits

Link Group ID 0xFFFFFFFF is reserved and indicates all data links in a TE link. All data links are members of Link Group 0xFFFFFFFF by default.

Channel_Status: 32 bits

The values for the Channel_Status field are defined in [LMP].

This Object is non-negotiable.

2.5. Alarm Management

Alarm management is an important feature of LMP-WDM because it can be used to suppress cascading and/or spurious alarms during normal connection procedures. For example, the OXC may know that a connection is ôupö, ôdownö, in a ôtestingö mode, or being deleted (ôdeletion-in-progressö). The OXC can use this information to inhibit alarm reporting from the OLS when the state of a connection changes in a controlled fashion.

Alarm management is controlled using the Active bit of the CHANNEL_STATUS object (see [LMP]).

In the following, we describe how the Active bit can be used in conjunction with the Admin Status object of [GMPLS] to manage alarms during graceful connection deletion.

Consider the network of Figure 3 where a wavelength LSP has been established using RSVP-GMPLS from OXC-A through OXC-B to OXC-C. To support graceful deletion of the LSP, the Deletion in Progress bit is set in the Admin Status object of a Path message that is transmitted from OXC-A through OXC-B to OXC-C. This bit indicates that ôlocal actions related to LSP teardown should be taken.ö As part of the local actions for LSP teardown, each OXC should notify itÆs neighboring OLS(s) that the data link is now deactive. For example, OXC-B should notify OLS-B1 and OLS-B2 that the link is deactive before forwarding the Path message to the next node. This ensures that when the connection is removed multiple alarms are not triggered at each of the line systems.

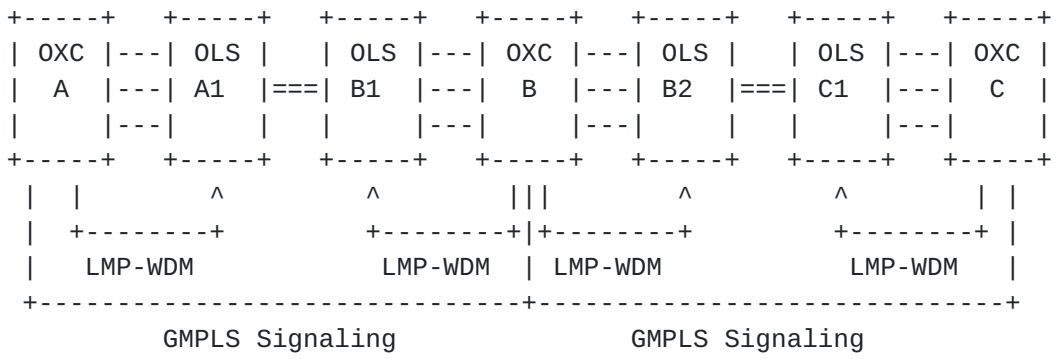


Figure 3: Alarm Management Example

2.6. Trace Monitoring

The trace monitoring features described in this section allow a PXC to do basic trace monitoring on circuits by using the capabilities on an attached OLS.

- An OLS Client may request the OLS to monitor a link for a specific pattern in the overhead using the TraceMonitorReq Message. An example of this overhead is the SONET Section Trace message transmitted in the J0 byte. If the actual trace message does not match the expected trace message, the OLS MUST report the mismatch condition.

- An OLS client may request the value of the current trace message on a given data link using the TraceReq Message.

2.6.1. TraceMonitor Message (MsgType = TBD)

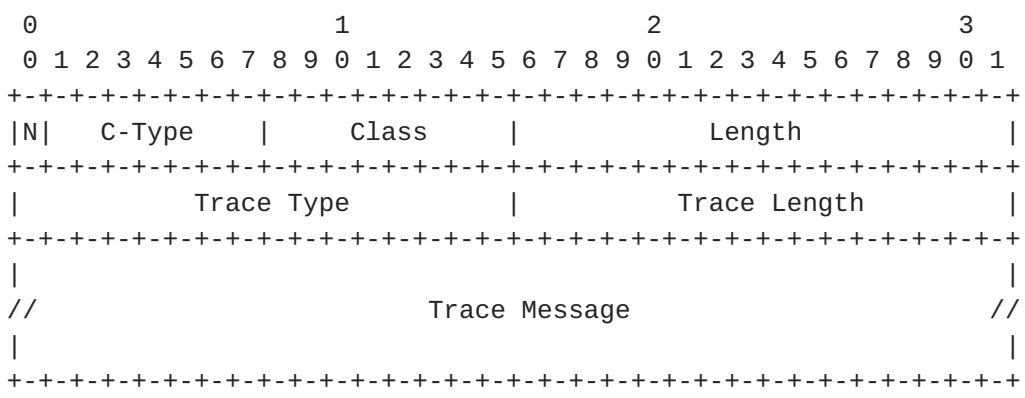
The TraceMonitor message is sent over the control channel and is used to request an OLS to monitor a data link for a specific trace value. An OLS MUST respond to a TraceMonitor message with either a TraceMonitorAck or TraceMonitorNack Message.

```
<TraceMonitor Message> ::= <Common Header> <MESSAGE_ID>
                  <LOCAL_INTERFACE_ID> <TRACE>
```

If supported by the hardware, traces of different types may be monitored simultaneously (e.g., Section and Path trace messages may exist simultaneously on the same data link).

2.6.1.1. TRACE Object

The format of the TRACE object is as follows (Class and C-Type to be assigned by IANA):



The Trace Object is non-negotiable.

Trace Type: 16 bits

The type of the trace message:

1 û SONET Section Trace (J0 Byte)
2 û SONET Path Trace (J1 Byte)
3 û SDH Section Trace (J0 Byte)
4 û SDH Path Trace (J1 Byte)

Other types TBD.

Trace Length: 16 bits

The Length in bytes of the trace message provided.

Trace Message:

Expected message. The valid length and value combinations are determined by the specific technology (e.g., SONET or SDH) and are beyond the scope of this document. The message MUST be padded with zeros to a 32-bit boundary, if necessary.

2.6.2. TraceMonitorAck Message (MsgType = TBD)

The TraceMonitorAck message is used to indicate that all of the Trace Objects in the TraceMonitor message have been received and processed correctly.

The format is as follows:
<TraceMonitorAck Message> ::= <Common Header> <MESSAGE_ID_ACK>

The MESSAGE_ID_ACK object is defined in [LMP].

2.6.3. TraceMonitorNack Message (MsgType = TBD)

The TraceMonitorNack message is used to indicate that the Trace

Object in the TraceMonitor message was not processed correctly. This could be because the trace monitoring requested is not supported or there was an error in the value.

The format is as follows:

<TraceMonitorNack Message> ::= <Common Header> <MESSAGE_ID_ACK>
<ERROR_CODE>

The MESSAGE_ID_ACK object is defined in [LMP].

The TraceMonitorNack message uses the ERROR_CODE C-Type,

2.6.3.1. ERROR_CODE Class

C-Type = 20 (see [LMP])

LMP-WDM defines the following new error code bit-values:

TBD1 = Unsupported Trace Type
TBD2 = Invalid Trace Message

All other values are Reserved.

Multiple bits may be set to indicate multiple errors.

This Object is non-negotiable.

2.6.4. TraceMismatch Message (MsgType = TBD)

The TraceMismatch message is sent over the control channel and is used to report a trace mismatch on a data link for which trace monitoring was requested.

A neighboring node that receives a TraceMismatch message MUST respond with a TraceMismatchAck message. The format is as follows:

<TraceMismatch Message> ::= <Common Header> <MESSAGE_ID>
<LOCAL_INTERFACE_ID> [<LOCAL_INTERFACE_ID> ...]

The LOCAL_INTERFACE_ID object is defined in [LMP]. The LOCAL_INTERFACE_ID in this message is the local Interface Id of the link that has a trace mismatch. A trace mismatch for multiple LOCAL_INTERFACE_ID's may be reported in the same message.

2.6.5. TraceMismatchAck Message (MsgType = TBD)

The TraceMismatchAck message is used to acknowledge receipt of a TraceMismatch message.

The format is as follows:
<TraceMismatchAck Message> ::= <Common Header> <MESSAGE_ID_ACK>

The MESSAGE_ID_ACK object is defined in [LMP] and must be copied from the TraceMismatch Message being acknowledged.

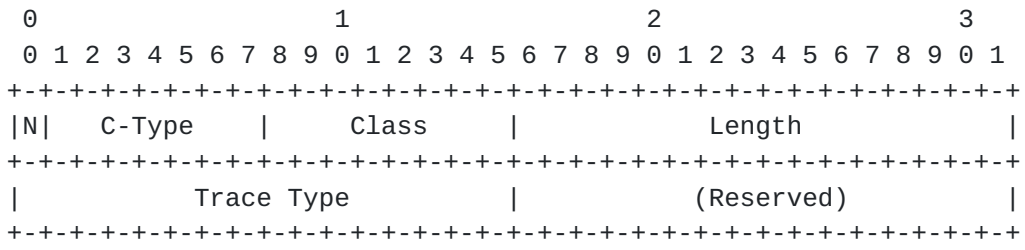
2.6.6. TraceReq Message (MsgType = TBD)

The TraceReq message is sent over the control channel and is used to request the current trace value of indicated data links.

A node that receives a TraceReq message MUST respond with a TraceReport message. The format is as follows:

```
<TraceReq Message> ::= <Common Header> <MESSAGE_ID>
                       <LOCAL_INTERFACE_ID> <TRACE REQ>
```

The format of the TRACE_REQ object is as follows:



Trace Type: Defined in Section 2.6.1.1.

2.6.7. TraceReport Message (MsgType = TBD)

The TraceReport message is sent over the control channel after receiving a TraceReq message.

```
<TraceReport Message> ::= <Common Header> <MESSAGE_ID_ACK> <TRACE>
```

The TraceReport message MUST include a TRACE Object (as described in Section 2.6.1.1) for the link requested.

2.6.8. TraceReqNak Message (MsgType = TBD)

The TraceReqNak message is sent over the control channel after receiving a TraceReq message.

```
<TraceReqNak Message> ::= <Common Header> <MESSAGE_ID_ACK>
                          <ERROR_CODE>
```

The TraceReqNak message MUST include an ERROR_CODE Object (as described in Section 2.6.3) for the link requested.

2.6.9. InsertTraceReq Message (MsgType = TBD)

The InsertTraceReq message is sent over the control channel and is used to request an OLS to send a specific trace message on a data link. An OLS MUST respond to a InsertTraceReq message with either a InsertTraceAck or InsertTraceNak Message.

```
<InsertTraceReq Message> ::= <Common Header> <MESSAGE_ID>
                             <LOCAL_INTERFACE_ID> <TRACE>
```

2.6.10. InsertTraceAck Message (MsgType = TBD)

The InsertTraceAck message is used to indicate that the TRACE Object in the InsertTrace message has been received and processed correctly.

The format is as follows:
<InsertTraceAck Message> ::= <Common Header> <MESSAGE_ID_ACK>

The MESSAGE_ID_ACK object is defined in [LMP].

2.6.11. InsertTraceNack Message (MsgType = TBD)

The InsertTraceNack message is used to indicate that the Trace Object in the InsertTrace message was not processed correctly. This could be because the trace monitoring requested is not supported or there was an error in the value.

The format is as follows:
<InsertTraceNack Message> ::= <Common Header> <MESSAGE_ID_ACK>
<ERROR_CODE>

The MESSAGE_ID_ACK object is defined in [LMP]. The ERROR_CODE Object usage is described in Section 2.6.3.1.

3. Security Considerations

General LMP security issues are discussed in [LMP]. As in [LMP], LMP-WDM exchanges may be authenticated using the Cryptographic authentication option. MD5 is currently the only message digest algorithm specified. The InsertTraceReq and TraceMonitor messages introduced in this document present an opportunity for an intruder to disrupt transmission. Authentication of messages is recommended if the control network itself is not secure.

4. Work Items

The following work items have been identified. They will be addressed in a future version of this draft:

1. Error messages may be needed in response to some of the defined messages.

2. More discussion on Trace Monitoring procedures is needed.

3. Provide description of procedures and interactions for running LMP and LMP-WDM on the same link. Include description of how control over link transparency works during the Verify procedure.

4. Determine whether some functions are optional and, if so, provide a capability negotiation mechanism.

5. References

[GMPLS] Berger, L., Ashwood-Smith, Peter, editors, "Generalized MPLS - Signaling Functional Description", Internet Draft, draft-ietf-mpls-generalized-signaling-02.txt, (work in progress), March 2001.

[Bra96] Bradner, S., "The Internet Standards Process -- Revision 3," BCP 9, RFC 2026, October 1996.

[DBC00] Drake, J., Blumenthal, D., Ceuppens, L., et al., "Interworking between Photonic (Optical) Switches and Transmission Systems over Optical Link Interface (OLI) using Extensions to LMP", OIF Contribution oif2000.254, (work in progress), November 2000.

[KRB00] Kompella, K., Rekhter, Y., Berger, L., "Link Bundling in MPLS Traffic Engineering," Internet Draft, draft-kompella-mpls-bundle-02.txt, (work in progress), July 2000.

[KRB00a] Kompella, K., Rekhter, Y., Banerjee, A., et al, "OSPF Extensions in Support of Generalized MPLS," Internet Draft, draft-kompella-ospf-extensions-00.txt, (work in progress), July 2000.

[LMP] Lang, J., Mitra, K., Drake, J., Kompella, K., Rekhter, Y., Berger, L., Saha, D., Basak, D., Sandick, H., Zinin, A., "Link Management Protocol (LMP)", Internet Draft, draft-ietf-ccamp-lmp-02.txt, (work in progress), July 2001.

[OLI] Fredette, A., Editor, "Optical Link Interface Requirements", Internet Draft, draft-many-oli-reqts-00.txt, (work in progress), June 2001.

[SDH] ITU-T G.707, "Network node interface for the synchronous digital hierarchy (SDH)", 1996.

[SONET] GR-253-CORE, "Synchronous Optical Network (SONET) Transport Systems: Common Generic Criteria", Telcordia Technologies, Issue 3, September 2000

[T.50] ITU-T T.50, "International Reference Alphabet (IRA) (formerly International Alphabet No. 5 or IA5) Information technology 7-bit coded character set for information interchange.", 1992.

6. Author's Addresses

Osama S. Aboul-Magd
Nortel Networks
P.O. Box 3511, Station C
Ottawa, Ontario, Canada
K1Y 4H7
Phone: 613-763-5827
email: osama@nortelnetworks.com

Stuart Brorson
Axiowave Networks
100 Nickerson Road
Marlborough, MA 01752
email: sdb@axiowave.com

Sudheer Dharanikota
Nayna Networks, Inc.
157 Topaz Drive,
Milpitas, CA 95035
email: sudheer@nayna.com

John Drake
Calient Networks
5853 Rue Ferrari
San Jose, CA 95138
email: jdrake@calient.net

David Drysdale
Data Connection Ltd
dmd@dataconnection.com

W. L. Edwards
iLambda Networks
Aspen, CO
email: texas@ilambda.com

Adrian Farrel (Movaz Networks)
Movaz Networks, Inc.
7926 Jones Branch Drive,
Suite 615
McLean, VA 22102
email: afarrel@movaz.com

Andre Fredette
PhotonEx Corporation
8C Preston Court
Bedford, MA 01730

Rohit Goyal
Axiowave Networks
100 Nickerson Road
Marlborough, MA 01752
email: rgoyal@axiowave.com

Hirokazu Ishimatsu
Japan Telecom
2-9-1 Hatchobori. Chuo-ku,
Tokyo, 104-0032 Japan
email: hirokazu@japan-
telecom.co.jp

Monika Jaeger
T-systems
Monika.Jaeger@t-systems.de

Ram Krishnan
Axiowave Networks
100 Nickerson Road
Marlborough, MA 01752
email: ram@axiowave.com

Jonathan P. Lang
Calient Networks
Court25 Castilian Drive
Goleta, CA 93117
email: jplang@calient.net

Raghu Mannam
Hitachi Telecom (USA), Inc.
rmannam@hitel.com

Eric Mannie
Ebone (GTS)
Terhulpsesteenweg 6A
1560 Hoeilaart
Belgium
Email: eric.mannie@gts.com

Dimitri Papadimitriou
Alcatel
Francis Wellesplein 1,
B-2018 Antwerpen, Belgium
email: dimitri.Papadimitriou
@alcatel.be

email: fredette@photonex.com

Jagan Shantigram
PhotonEx Corporation
8C Preston
Bedford, MA 01730
email: jagan@photonex.com

Yong Xue
UUNET/WorldCom
22001 Loudoun County Parkway
Ashburn, VA 20148
email: yxue@uu.net

Ed Snyder
PhotonEx Corporation
8C Preston Court
Bedford, MA 01730
email: esnyder@photonex.com

Lucy Yong
Williams Communications
2 East First Street
Tulsa, OK 74172
lucy.yong@wilcom.com

George Swallow
Cisco Systems, Inc.
250 Apollo Drive
Chelmsford, MA 01824
Email: swallow@cisco.com

John Yu
Zaffire, Inc
2630 Orchard Parkway
San Jose, CA 95134
email: jzyu@zaffire.com

Gopala Tumuluri
Calient Networks
5853 Rue Ferrari
San Jose, CA 95138
email: krishna@calient.net