

Link Management in Next Generation Optical Networks

Zhiyu Zhou Caixia Chi
15/F Aerospace Great Wall Building
No. 30 Hai Dian Nan Lu
Bell Labs Research China
Beijing 100080, China
{zhiyuzhou, chic}@lucent.com

Abstract

With the development of dense wavelength-division multiplexing (DWDM) and micro electro-mechanical system (MEMS) technologies, optical switch becomes the key device in today's optical networks. Since the optical switch has a large number of data links, a local link management protocol is necessary to control and manage data links to discover port associations for light-path computing automatically, monitor the health of data links, and detect and locate link faults for protection or restoration. This paper investigates and compares several mechanisms for link management in both control and data planes, with special focus on the neighbor discovery approaches. Since link management protocol (LMP) is being standardized as a part of generalized multiprotocol label switching (GMPLS), the related issues are also addressed in this paper.

1. Introduction

The explosive growth of the Internet and related services is strengthening the demand for more bandwidth than what could be provided by traditional data networks. With features of high capacity, ample bandwidth, and economic feasibility, optical network is becoming an ideal solution for the next generation networks. Due to the extraordinary technology innovations and service-driving demands over the last two decades, the optical layer has moved from providing the simple point-to-point optical fiber transmission to enabling an intelligent network [1]. The distinct manifest of an intelligent optical network is a control plane separated from the data plane to provide automatic wavelength provisioning, network resource management, and light-path protection and restoration, as shown in Fig. 1. Here, an optical data plane enables the transmission of traffic in a robust, scalable, and cost-effective manner and consists of optical switches as its key elements, as well as DWDM transmission system that can handle long-haul spans. While the control plane provides the intelligence to light-path protection or restoration, routing, and signaling.

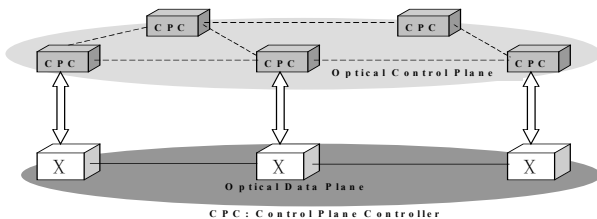


Fig. 1: Control and data planes in an optical network.

A flexible and scalable control plane should have key functions such as routing, signaling, protection/restoration, quality of service, and automatic operation process [2]. A standardized control plane for optical networks gives lots of benefits to carriers and service providers. Therefore, there are several protocols being pushed by various standard organizations including IETF, OIF, and ITU-T, such as GMPLS, Optical User Network Interface (O-UNI), ATM Private Network-to-Network Interface (PNNI) extension, or a completely new one. Since none of those protocols is completely finalized so far, the gradual steps towards the optical signaling network include: enhancing Operational Support Systems and Network Management Systems (OSS/NMS) to allow automatic service provisioning (point-and-click) of circuits, connecting edge network elements (ATM, IP, etc.) to the core optical network by O-UNI, and implementing a unified control plane. The search for an optical control plane was made easier by the rise and rapid development of MPLS that has already defined a control plane separated from the data-forwarding plane. Moreover, MPLS provides circuit switching that is natural for wavelength setup and teardown, and the basic idea of label switching could be used while a lambda is regarded as a label in optical networks. As a result, GMPLS is being standardized by IETF to provide the necessary bridge between IP and photonic layers to allow the interoperable and scalable parallel growth in IP and photonic dimensions [3][4].

GMPLS needs to extend MPLS to address arising issues related to the optical domain by encompassing not only the enhancements to the routing and signaling parts, but also a new specialized protocol, LMP (specialized to GMPLS-LMP in this paper) to control and manage data links in optical networks. In today's optical networks, DWDM optical products are deployed, so there are a very large number of parallel links between two adjacent nodes. This makes manual configuration, control and management, and link state information flooding impractical. Moreover, since the control channels between two adjacent nodes are no longer required to use the same physical medium as the data-bearing links in optical networks, it is necessary to dedicate a management protocol to handle link provisioning, link health maintenance, and fault tolerance in both control and data planes.

This paper introduces and compares several mechanisms for link management protocols from both control and data planes, classifies neighbor discovery schemes, and brings

out some issues related to LMP for its standardization. The rest of this paper is organized as follows. Section 2 gives an overview of link management functions. Various schemes used for link management are described in control and data planes in Section 3 and 4 respectively. Since LMP is being standardized under the umbrella of GMPLS, some further issues that need to be taken into consideration are presented in Section 5. Finally, we conclude the current work and suggest some future works.

2. Link Management Functions

As we said above, both the separated control plane from the data plane and large number of links between any pair of nodes present new challenges to the link management. Besides that, it is also more complex in today's optical network where an OXC usually connects to optical line systems (OLSs) to deliver high bandwidth over long distance. Those OLSs are transparent to attached OXCs as link management sets up neighbor associations for path computation and link protection or restoration. However, more and more OLSs manufactured today could provide fault notification to the neighboring OXC. Therefore link management could also be used between one OXC and its attached OLS. Furthermore, at the edge of the core network, the edge OXCs always attach to some client devices such as IP routers or ATM switches. Link management could also be localized between the edge OXC and client devices to provide automatic neighbor discovery, service discovery, and address registration. Therefore a unified link management in an optical network could be illustrated in Fig. 2.

To control and manage totally different links connecting optical devices, link management should have basic functions such as link provisioning, link health monitoring, and link fault management. Since optical networks allow the control plane to be physically decoupled from the data plane, link management should provide the control and management not only for component data links, but also for control channels. Moreover, the health of links in two different planes should not be correlated with each other. The following sections describe link management from three main functions of neighbor discovery, link health monitoring, and fault management in both the control and the data planes.

3. Link Management in the Control Plane

The control plane is physically decoupled from the data plane in the core optical network. It provides a data communication network for signaling such as RSVP-TE or CR-LDP, routing such as OSPF-TE, and other link management functions such as link property correlation. Therefore we also need to provide link management inside the control plane. However, the implementation of the control plane network is totally different from the data plane, e.g., a dedicated LAN. It is easier to manage the links in the control plane than optical links in the data plane. The key issue here is the reliability.

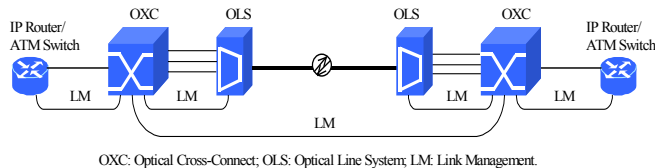


Fig. 2. Link management for different devices in an optical network.

To control and manage a data link, controllers attached to the end devices should build neighbor associations in the control plane. This control channel connection could be manually configured with IP entity of each other to setup UDP or TCP sessions. In LMP, more functions, like control channel parameter negotiation, are provided by exchanging configuration messages.

After neighbor associations are established, the network health could be monitored by fast keep-alive mechanism, i.e., the hello protocol. When a connection failure is detected, the fault can be tolerated by switching traffic to other control channels as in LMP[6]. Since multiple control channels might be different only logically but sharing the same physical media, for more reliability, connections could be protected by MPLS label switched path (LSP) protection, in which two physically disjoint LSPs are established between the same pair of nodes. Messages are transmitted over both simultaneously. If one of them breaks down, with the other one as backup, there won't be any interruption. At the same time a new LSP is computed.

4. Link Management in the Data Plane

In the data plane, link management is driven by making the management of a large number of optical links feasible. From the functional perspective, link management in the data plane should encompass automatic neighbor discovery, link health maintenance, and fault tolerance. Before the approaches meeting those three functions are described in detail, it is necessary to introduce the port properties of OXC.

Compared to traditional electrical switches handling electronic signals, an optical switch, on the other hand, works with light. It directs a single wavelength or a range of wavelengths from an input port to an output port. When a port has optical-electrical-optical (OEO) conversion device, referred to as an opaque port; it can retrieve information from the received modulated optical signal. Otherwise, the transparent ports can only detect the transition of incoming optical signals from presence to absence or vice versa. It cannot deal with the information embedded in the signal.

Based on the location of the OEO device, optical cross-connects can be classified into OEO switches, all optical switches, and OEO-O-OEO switches. In an OEO switch, the input/output modules are transparent, but receivers turn the photons into electrons for their journeys over an electronic backplane. At the output module, the electrons are converted back into photons. This is how synchronous

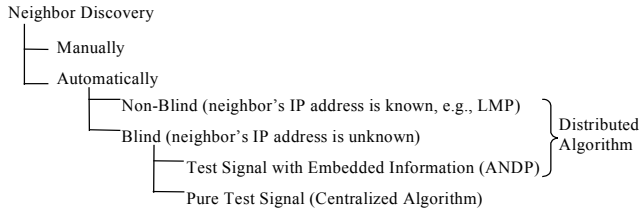


Fig. 3. Classification of neighbor discovery algorithm

optical network/synchronous digital hierarchy (SONET/SDH) cross-connects, FDDI switches, ATM switches, Ethernet switches (with fiber optical interfaces), and routers (with FDDI, ATM, or POS interfaces) have worked for more than a decade. The electronic backplane has limits on both capacity and scalability, but it is easy to access the information. The all-optical switch is an all-analog device in recent development, where both the input/output modules and the backplane are transparent. All-optical switch provides high capacity and cost reduction, but no visibility of bit errors rate. The compromise of the former two architectures, OEO-O-OEO switches place the electronic conversion in the input/output module, and use an optical backplane. Since the OEO conversion module is expensive, most ports in an OEO-O-OEO switch are transparent. When it is needed to retrieve information from the optical signal, the test port with OEO conversion can be connected to the transparent ports to complete this function. Most state-of-the-art optical switches fall into this category.

4.1 Neighbor Discovery

In an optical network, neighboring OXCs are connected through OLSs supporting multiple wavelengths. Each wavelength is either originated from or terminated at a specific port of an OXC. For point-and-click provisioning, the management system of an OXC needs to have the knowledge of the number of available wavelengths in each direction, between it and every neighbor nodes, and the ports associated with them. This information can be provided through a discovery mechanism initiated manually or automatically.

In the manual mode, the user usually specifies the IP address, shelf, slot, and port of the neighboring nodes via commands. This is tedious and error-prone when processing a large number of links.

Automatic neighbor discovery mechanisms can be divided into 'non-blind' and 'blind' based on whether the other node's IP address is known or not. Both of the adjacent nodes know the IP address of each other in the 'non-blind' scheme, such as LMP. Therefore the neighbor discovery is actually port discovery and negotiation messages are exchanged before delivering a real test signal. In the 'blind' discovery, the node and port information is modulated into a test signal, and the port association is established based on the information retrieved from the test signal. Automatic Neighbor Discovery Protocol (ANDP) [5] is such an example. From the perspective of architecture, those mechanisms can also

be classified into centralized algorithms and distributed ones, as shown in Fig. 3.

4.1.1 Centralized Algorithm

In the centralized algorithm, neighbor discovery process is controlled by a leader which is an optical device elected automatically or designated manually. All the messages related with the leader identification, recognition request, and port association are carried over the control communication network in the control plane, while only no information-embedded light is sent to test the link connectivity in the data plane between two adjacent nodes with the grant from the leader.

Fig. 4 shows a neighbor discovery process with node 1 as designated leader. Initially, each node checks the leader identity flag configured locally. If not a leader itself, it periodically broadcasts the leader-acquiring message. Upon receiving this message, the leader sends back an acknowledgement to that node. Once the leader is known by all nodes, nodes requiring neighbor discovery send the leader all their unrecognized output ports via discovery request messages. The leader puts all to-be-discovered ports from different nodes including its own into a FIFO queue. To process each port, the leader sends a discovery grant message to the owner of the port. Upon receiving such a grant message, the owner sends light through this unrecognized output port. Once an input port connected to this output port detects the test signal, it will send a discovery detected message to the leader. Receiving such message, the leader sends the port association to the original request node as well as stops the discovery for this unrecognized port. Finally, it sends the same information to the node owning the corresponding input port to complete the procedure. This procedure is repeated for each port in the to-be-discovered port queue.

Although the port just simply sends light without embedded information in the data plane, the burden to the control plane makes the centralized algorithm too complicated and time-consuming to be employed. Compared to it, distributed algorithms are much more efficient and easier for implementation.

4.1.2 ANDP

ANDP is a distributed automatic neighbor discovery

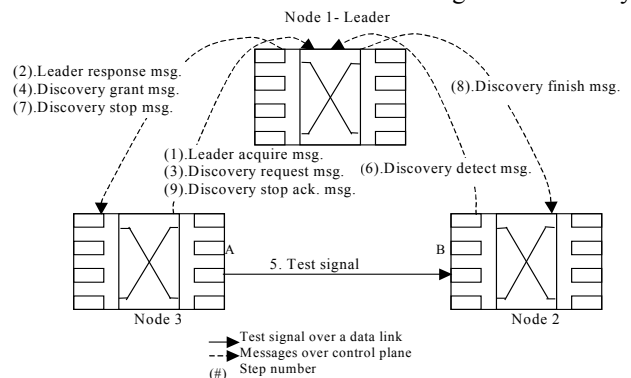


Fig. 4. Neighbor discovery with the centralized algorithm.

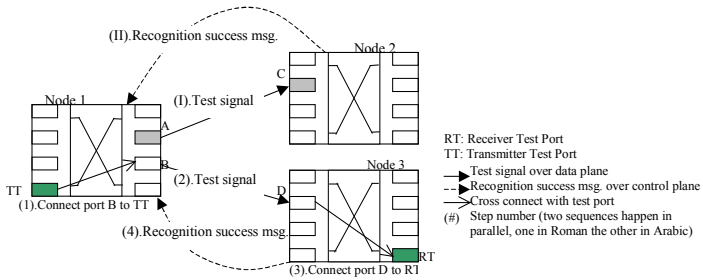


Fig. 5. Automatic neighbor discovery with ANDP.

protocol proposed by Bell Labs Research China. It was implemented and integrated into the optical network management protocol (ONMP) [5]. In ANDP, there is no leader to coordinate the discovery procedure. Each node maintains its own to-be-discovered port list. The port recognition process is initiated by any node that has a non-empty list of unrecognized output ports. It sends a test signal from an unrecognized output port to its downstream node over the data link. The test signal contains the information about the port attributes. If the input port connecting to that output port is opaque and detects the signal, the embedded information is extracted from the test signal directly, otherwise a test port is used to extract the information embedded in the test signal. Once the downstream node gets the necessary information, a recognition success message is sent back to the upstream node over the control plane. From this message, the upstream node obtains the port association as well as the information about its neighboring downstream node.

Fig. 5 shows an example of ANDP with three nodes. Node 1 has two unrecognized output ports: port A and B that is opaque and transparent respectively. Input port C of node 2 is opaque while D of node 3 is transparent. As a distributed protocol, the discovery for port A and B are performed in parallel. The association between port A and port C is much easier since both of them are opaque. While since port B and D are transparent, a transmitter test port (TT) in node 1 and a receiver test port (RT) in node 3 must be called up.

In ANDP, the transmitter and receiver test port are required to embed and extract information for transparent ports. The information embedded in a test signal should include node identifier and port identifier, since it has to be informed to the neighboring node for exchanging control and management messages. ANDP could populate the SONET J0 overhead byte with the information in a test signal.

4.1.3 LMP

As one of the main parts of GMPLS, LMP can be deployed onto different devices include photonic switches, optical cross-connects, and routers in the optical network. In [6], neighbor discovery function is divided into two parts: link property correlation and link connectivity verification.

Link property correlation is to aggregate multiple data links into a traffic engineering (TE) link and to correlate

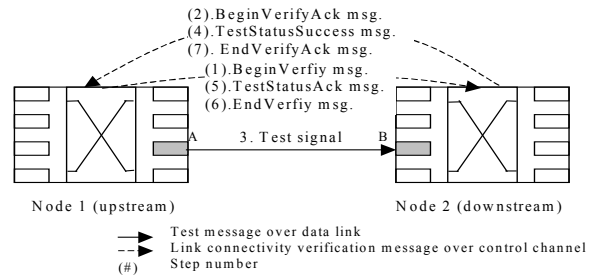


Fig. 6. Link connectivity verification of LMP.

the properties of that link. The messages exchanged for link property correlation include LinkSummary, LinkSummaryAck, and LinkSummaryNack.

If there is only one plane for both control and data traffic, the link property correlation can also be used to ensure the consistency of the link information about TE and data bearing links on both sides. However, that is not the case in the data plane separated from the control plane. Therefore link connectivity verification is particularly useful in this situation to verify the physical connectivity of data-bearing links between nodes and to exchange the interface identifiers. In LMP, this procedure is initiated by a BeginVerify message delivered over control channel. Once a BeginVerifyAck is received in response, the test signal are transmitted over the data link, and then TestStatusSuccess and TestStatusAck are exchanged. If there is more than one data link to be processed in link connectivity verification, the exchange of Test, TestStatusSuccess and TestStatusAck needs to be repeated for every link. EndVerify and EndVerifyAck are exchanged over the control channel at the end of the procedure. The link connectivity verification for a single data link is illustrated in Fig. 6. The test signal used to verify the data link includes only the local interface identifier and the verify identifier, while other information such as encoding type, transmission rate of test signal etc., is included in the BeginVerify message. Two neighboring nodes need to know IP address of each other to transfer BeginVerify message. Compared to ANDP, LMP adds much more information to control messages and increases the number of control messages required for the neighbor discovery.

4.1.4 Neighbor Discovery Extension for Other Devices

Neighbor discovery is performed not only between adjacent OXCs [6], but also between an edge OXC and a client device [7] or between an OXC and an OLS attached to it [8]. These extensions usually make the OXC the active node while the other device the passive node. The port associations between two adjacent OXCs are to provide necessary information for light-path computation and connection setup, while the port associations between the OXC and its attached OLS are also established for fault localization. Moreover, it's also necessary to set up the port relationships between the OXC and its interconnected client device to provide the client service discovery and address registration. To achieve a unified neighbor

discovery, the approaches described above can be used separately or combined in the entire optical network.

4.2 Link Health Monitoring

Once the neighbor discovery procedure is completed, a monitor procedure should be provided to validate the available path and the neighbor information is consistent with what was discovered earlier. The different neighbor discovery approaches described above can be employed for such monitor procedure, especially for the OXCs whose attached OLSs could not provide any monitor mechanisms, i.e., transparent to OXCs. In this case, OLS just passes any signals to the OXC connecting to it and has no communication to the neighboring OXC in the control plane. Such monitor procedure could be done periodically, e.g., once or a couple of times in a day.

However, some of today's OLSs provide intelligence and can generate some kind of keep-alive signal, typically a SONET alarm indication signal (AIS). Such signal can detect the failure in the transport system. Once the neighbor discovery procedure is completed, the keep-alive signal is generated by the OLS to monitor the health of data links when there is no client traffic. If a failed data links has been detected, it will not be used for client service traffic or will be restored. Since this low-level mechanism is employed to check the health of data links, the original neighbor discovery used for monitor procedure could be enhanced, but in some case adjustments are needed. The centralized algorithm and the link connection verification of LMP can be used directly. However, when using ANDP, the upstream OXC should inform the downstream OXC the start of the monitor procedure, because the transparent port could not tell the test signal from the keep-alive signal. Therefore a specialized message is needed to inform the start of the monitor procedure.

4.3 Fault Management

4.3.1 Fault Management among OXCs

Fault management is another important function of link management. It provides fault detection, localization, and notification and also trigger fast path or link protection/restoration when failure occurs. The transparent ports located in OXC could only monitor the presence or the absence of signal, usually known as detection of signal (DOS) and loss of signal (LOS). Therefore the fault detection relies solely on the DOS or LOS in the input port.

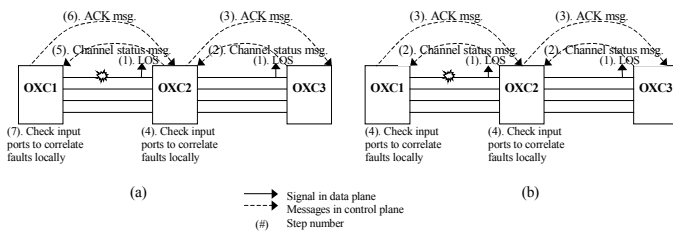


Fig. 7. Fault localization: (a) serial process; (b) parallel

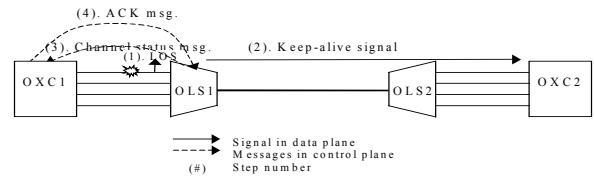


Fig. 8. Fault management extension between OXC and OLS.

The implementation of fault management depends on what kind of protection/restoration scheme is adopted. In path protection/restoration, the ingress or egress node could initiate a quick reroute. Therefore, once the fault is detected by all downstream nodes because of LOS, the egress node could trigger the protection or send a message in the control plane to inform the ingress node to initiate reroute.

If link protection/restoration is provided, the fault must be localized to the failed link before reroute. The process for fault localization could be implemented serially or in parallel, as shown in Fig. 7. In serial process, the egress node initiates the fault localization process by sending a channel status message to its upstream node. Once the upstream node receives the message, it will send an acknowledgement message back, and check its input ports to determine the location of faults. If no fault is determined to occur between them, the upstream node will issue a channel status message to the node further upstream to continue the fault localization process. However, in parallel process, all downstream nodes detecting LOS initiate fault localization procedure simultaneously. LMP employs the parallel approach, which can speed up the response to the fault. The serial approach can reduce the burst messaging traffic in the network, but needs more time to localize the fault.

4.4 Fault Management Extension for OLSs

The fault management among OXCs can only localize the failure between two nodes and takes much time to do it. Today's OLS can detect fault, localize it, and notify its attached OXC for quick restoration. The keep-alive signal generated by OLSs not only monitors the health of data links, but also enables the fault detection when failures occur.

The OLS may have its own fault localization process which can locate the fault much more specifically than the OXC does. If OLS detects LOS at the input port, it will generate keep-alive signal downstream, and communicate with the OXC it attached to in the control plane, the failure will be localized more quickly and the restoration process can be initiated. Therefore, the fault management extension for OLS must allow it to notify its attached OXC when it detects a failure. The communication in the control plane could provide such quick notification function. Like the neighbor discovery function, this extension runs on the OLS and is independent of the end-to-end fault management of OXCs. But the two different processes could be combined to localize faults while avoiding

unnecessary interaction. An example of OLS fault management is shown in Fig. 8.

4.4.1 LMP Issues

LMP is proposed as a part of GMPLS to control and manage TE links because there are a large number of data links between two adjacent OXCs. It attempts to cover both control and data planes for neighbor discovery, link health maintenance, and fault management. Moreover, it is also extended to cover other devices such as OLSs, IP routers, and ATM switches. Therefore, it needs to take all kinds of scenarios into consideration. So far the proposal is a standard-track draft, many issues should be clarified in the final standard.

In LMP, if two adjacent nodes need to initiate any process, they must know the IP address of each other. In the draft, this is manually configured. To support auto-discovery of control channel endpoint addresses, a bootstrap procedure [9] is proposed to multicast a new message with the node IP address periodically before the whole link management process is initiated.

In the link connectivity verification, multiple data links are tested serially, i.e., only after a TestStatusSuccess message is received, LMP will process the next data link. This is time-consuming when there are lots of data links to be tested. If many ports are opaque or multiple testports presented, the test message sending does not depend on the test port availability and all data links can be verified in parallel.

Since LMP is extended to run between an OXC and its attached OLS as LMP-WDM, the OXC would run two different link management sessions for OXC-to-OXC and OXC-to-OLS. These two sessions will bring up interactions. Although the data link connectivity verification of OXC-to-OXC and OXC-to-OLS are independent of each other, the usage of the OXC-to-OLS data links are overlapped in both of the verification procedures. Therefore the OLS should know when it should be transparent to the test signal and when it should extract the test signal for OXC-to-OLS. Moreover, alarms caused by the verification procedure need be suppressed. For OXC-to-OLS verification, the OLS can be set to suppress unnecessary alarms upon receiving BeginVerify message. But for OXC-to-OXC, it has to use the control plane communication between the OXC and its OLS to inform the OLS to suppress alarms when the test signal is removed.

In a combined LMP and LMP-WDM environment, a fault can be detected by both OXCs and OLSs. Both of them will start their own fault localization to locate the fault. Although the fault can be localized more specifically in this case, the OXC would receive notifications from both its OXC peer and its attached OLS. It needs to be able to determine and correlate fault correctly in this situation.

5. Conclusions

In this paper, we show that a dedicated control plane is necessary to provide automatic wavelength provisioning,

network resource management, and light-path protection and restoration in optical networks. From the functional perspective, we have discussed many mechanisms to be implemented for link management in optical networks. The interconnection of different devices requires a standard. LMP proposed by IETF covers all the functions and is promising to be the final standard as part of GMPLS. To encompass wavelength division multiplexing devices, LMP is also extended as LMP-WDM. However, in its implementation, some issues need to be addressed, especially when both LMP and LMP-WDM run simultaneously in the same network. The order of neighbor discovery needs to be taken into consideration carefully to suppress unnecessary alarms caused by signals. Moreover, the properties of different devices also need further consideration. Since there are other protocols and schemes, LMP needs to prove essential to become a standard.

6. References

- [1] Rajiv Ramaswami, "Optical Fiber Communication: From Transmission to Networking", *IEEE Communications Magazine 50th Anniversary Commemorative Issue*, vol. 40, no. 5, pp. 138-147, May 2002.
- [2] Chunsheng Xin, Yinghua Ye, Ti-Shiang Wang, Sudhir Dixit, Chunming Qiao, and Myungsik Yoo, "On an IP-Centric Optical Control Plane", *IEEE Communications Magazine*, vol. 39, no. 9, pp. 88-93, Sept. 2001.
- [3] Ayan Banerjee, John Drake, J. Lang, Brad Turner, K. Kompella, and Y. Rekhter, "Generalized Multiprotocol Label Switching: An Overview of Routing and Management Enhancements", *IEEE Communications Magazine*, vol. 39, no. 1, pp. 144-150, Jan. 2001.
- [4] Ayan Banerjee, John Drake, J. Lang, Brad Turner, Daniel Awduche, K. Kompella, and Y. Rekhter, "Generalized Multiprotocol Label Switching: An Overview of Signaling Enhancements and Recovery Techniques", *IEEE Communications Magazine*, vol. 39, no. 7, pp. 144-151, July 2001.
- [5] Caixia Chi *et al.*, "Automatic Neighbor Discovery Protocol for Optical Networks", *Proceedings of APOC*, Nov. 2001.
- [6] Johnathan Lang, "Link Management Protocol (LMP)", *Internet draft, draft-ietf-ccamp-lmp-0.5.txt*, Aug. 2002, work in progress.
- [7] Osama Aboul-Magd *et al.*, "Implementation Agreement OIF-UNI-1.0", *OIF*, Oct. 2001.
- [8] A. Fredette *et al.*, "Link Management Protocol (LMP) for DWDM Optical Line Systems", *Internet draft, draft-fredette-lmp-0.3.txt*, Nov. 2001, work in progress.
- [9] J. Lang, J. Drake, and D. Papadimitriou, "Control Channel Bootstrap for Link Management Protocol", *Internet draft, draft-lang-ccamp-lmp-bootstrap-02.txt*, Dec. 2002, work in progress.