GMPLS-Based Control Plane for Optical Networks: Early Implementation Experience

Hang Liu, Dimitrios Pendarakis, Nooshin Komaee, Debanjan Saha
Tellium, Inc., 2 Crescent Place, Oceanport, NJ 07757

ABSTRACT

Generalized Multi-Protocol Label Switching (GMPLS) extends MPLS signaling and Internet routing protocols to provide a scalable, interoperable, distributed control plane, which is applicable to multiple network technologies such as optical cross connects (OXCs), photonic switches, IP routers, ATM switches, SONET and DWDM systems. It is intended to facilitate automatic service provisioning and dynamic neighbor and topology discovery across multi-vendor intelligent transport networks, as well as their clients. Efforts to standardize such a distributed common control plane have reached various stages in several bodies such as the IETF, ITU and OIF. This paper describes the design considerations and architecture of a GMPLS-based control plane that we have prototyped for core optical networks. Functional components of GMPLS such as signaling and routing are integrated in this architecture with an application layer controller module. Various requirements including bandwidth, network protection and survivability, traffic engineering, optimal utilization of network resources, and etc. are taken into consideration during path computation and provisioning. Initial experiments with our prototype demonstrate the feasibility and main benefits of GMPLS as a distributed control plane for core optical networks. In addition to such feasibility results, actual adoption and deployment of GMPLS as a common control plane for intelligent transport networks will depend on the successful completion of relevant standardization activities, extensive interoperability testing as well as the strengthening of appropriate business drivers.

Keywords: generalized MPLS, RSVP-TE, OSPF-TE, intelligent optical networks, distributed control plane

1. INTRODUCTION

Multiprotocol Label Switching (MPLS) integrates a label-based forwarding mechanism with the IP routing and control paradigm of packet data networks. A packet is classified and assigned a short fixed-length label as it enters an MPLS network. Then, the packet with the attached label is forwarded along a Label Switched Path (LSP) in the MPLS network. At each node across the path, the label is used to make forwarding decision. Each node strips off the existing label and applies a new label that instructs the next hop how to forward the packet. Label mapping needs to be set up at each node in advance in order to allow correct forwarding of packets along an LSP. A number of signaling protocols such as LDP[1], CR-LDP[2] and RSVP-TE[3] can be used to establish the label mapping and, consequently, the LSP. MPLS offers advantages and flexibility in the deployment of new applications, including traffic engineering (TE), support for differentiated quality of service, scalable Virtual Private Networks (VPN), and simplified integration of IP and other networking technologies (e.g. ATM). MPLS is currently being deployed by telecommunications carriers and Internet service providers.

The original MPLS framework consists of a data plane for packet forwarding and a control plane for LSP establishment. Generalized MPLS[4, 5] extends the MPLS control plane to support other data transport technologies. In addition to Packet-Switch Capable (PSC) interfaces, it defines three new interfaces, which correspond to three classes of transport technologies. These new interfaces are Time-Division Multiplex (TDM) Capable, Lambda (wavelength or waveband) Switch Capable (LSC), and Fiber-Switch Capable (FSC). GMPLS focuses mainly on the control plane for these new technologies. In particular, GMPLS reuses MPLS signaling and routing protocols with appropriate extensions and modifications based on generalized interface requirements. The suite of GMPLS protocols such as distributed signaling, routing, and link management is under standardization process in the Internet Engineering Task Force (IETF).

GMPLS enables a unified control plane that can be applied to a wide range of network equipment with different transport capabilities, such as IP routers, ATM switches, SONET/SDH cross-connects, optical cross-connects, photonic cross-connects, etc. This not only improves interoperability in a multi-vendor network with the same type of equipment but also provides seamless internetworking between different types of network elements. GMPLS provides the ability to dynamically provision, protect, and restore end-to-end LSPs across all above types of transport equipment, within and across multiple layers, eliminating or reducing the need for separate network control planes. For example, an LSP can be established between the PSC interfaces of two routers. This LSP can be nested into a SDH/SONET LSP that starts and ends on TDM interfaces. The SDH/SONET LSP, in turn, can be nested into a wavelength LSP between the LSC interfaces of optical cross-connects. The wavelength LSP can then be nested into a fiber LSP between the FSC interfaces of the photonic cross-connects. The application of distributed control improves network scalability and, since no single point failure exists, it presents advantages. Furthermore, in the case of multiple concurrent failures, the ability of every node to participate in routing protocols and perform path computation and provisioning may potentially provide better recovery characteristics.

Note that unlike MPLS, GMPLS allows de-coupling the control channel and data channels. GMPLS requires control entities to be able to communicate for the purpose of routing, signaling, and neighbor discovery and link management. However the interface over which control messages are exchanged may not be the same as the interface over which data flows. This offers enhanced flexibility in the deployment of new network control capabilities and applications.

We have developed a GMPLS-based control plane prototype for core optical switching networks. The prototype network is used to study the feasibility of GMPLS as a distributed control plane for provisioning, routing and neighbor discovery, as well as restoration of optically switched paths (LSPs) in a core optical network. Various requirements (traffic engineering, protection and restoration, bandwidth, etc.) have been taken into consideration during the LSP setup to optimize the network utilization. Another objective of the prototype is to study the different tradeoffs associated with GMPLS-based distributed control and to identify areas where the existing standardization proposals need to be expanded or improved.

This paper describes the design considerations, functionality and architecture of the GMPLS-based control plane software and prototype network. It is organized as follows. Section 2 discusses the GMPLS control plane functionality requirements and our development approach. In section 3, we describe the software architecture, the main components of the GMPLS control plane and the interfaces between the components. Section 4 presents the experimental network on which GMPLS is implemented. Finally, conclusions are presented in section 5.

2. GMPLS CONTROL PLANE DEVELOPMENT APPROACH

The main components of a GMPLS-based distributed control plane are network topology and link state dissemination, neighbor discovery and link management, path computation and signaling. The GMPLS control plane software development combined a top-down functionality analysis approach with a bottom-up architecture approach. Functionality analysis focused on deciding on standardized signaling and routing protocols, identifying feasible path computation algorithms as well as efficient methods to bundle unnumbered links based on optical network requirements. In this respect, RSVP-TE[6, 7] and OSPF-TE[8, 9,10] with GMPLS extensions were chosen as signaling and routing protocols, respectively, in our implementation.

A path computation module based on the constraint-based K-shortest path algorithm is employed. In order to reduce the amount of routing information distributed and to improve routing scalability, GMPLS provides the capability for links with similar characteristics to be advertised as a single *link bundle*, also known as *traffic engineering link*. Given that a core optical switch typically contains a large number of ports, there may be multiple links with similar characteristics between two nodes. Our implementation has defined an automated way to bundle links based on their characteristics such as speed and Shared Risk Link Group (SRLG).

The architecture bottom-up approach resulted in the development of several software components. In order to speed up the development, a commercial MPLS software package was used as the basis. This package was appropriately modified and extended to provide an extensible framework for GMPLS signaling and routing components. Our approach also led to the development and integration of new algorithms and software components specifically crafted for path computation and other functionality of core optical network applications. A prototype optical network has been developed. In this prototype network, GMPLS software components were integrated and tested for algorithm, protocol, and interface verification.

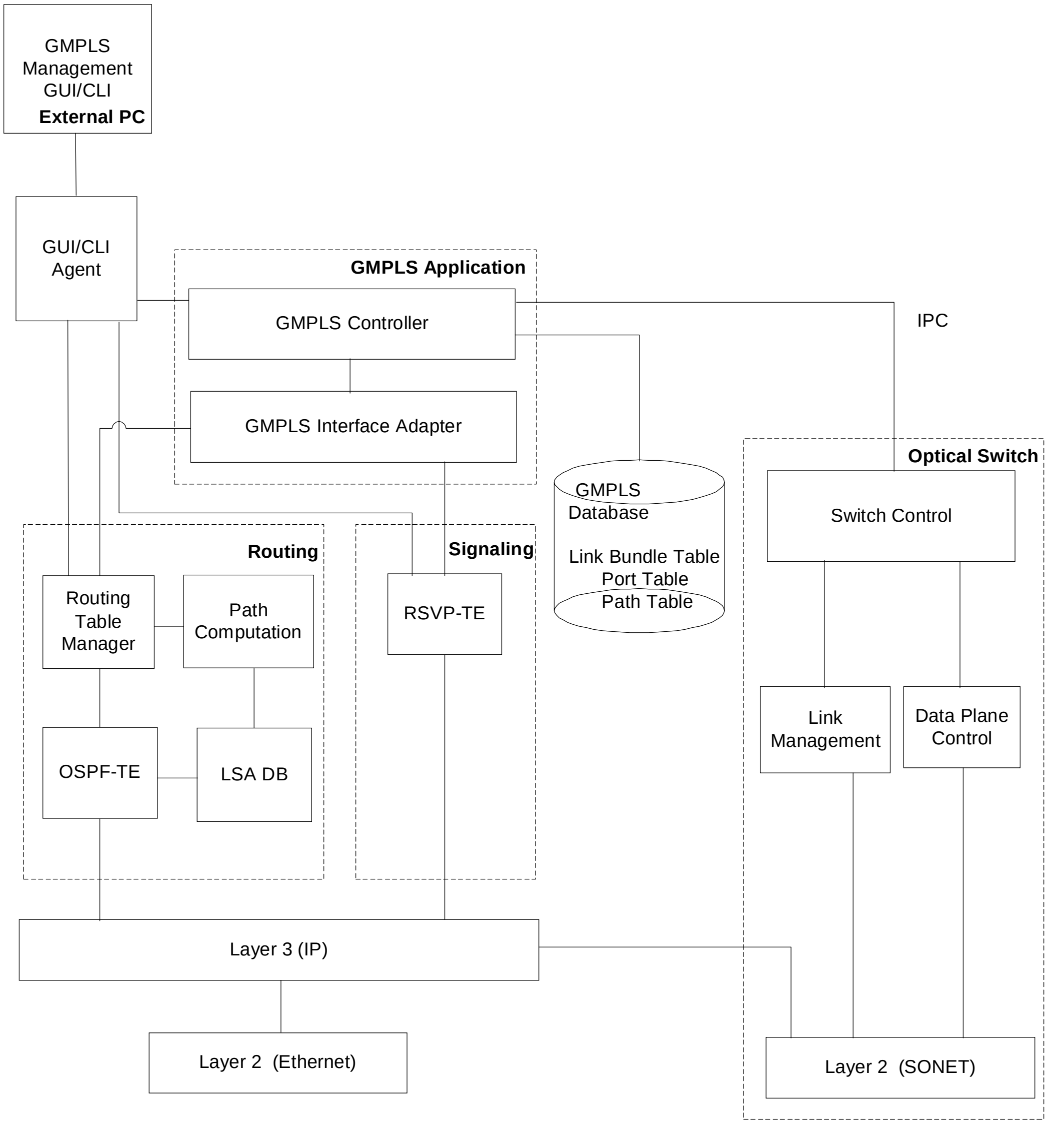

Figure 1: GMPLS control plane architecture.

3. GMPLS CONTROL PLANE SOFTWARE ARCHITECTURE

This section describes the architecture, key functions and interfaces of the main software components of the distributed, GMPLS-based control plane. As shown in Figure 1, the GMPLS control plane software consists of application component, signaling, routing, neighbor discovery and link management. A Graphical User Interface (GUI)/Command Line Interface (CLI) agent is also developed to provide an interface to an external console as well as EMS/NMS system for configuration, provisioning and display. Additionally, a GUI-based application has been developed to manage the distributed control plane and the prototype network. It is a Java-based application that runs on a host PC and provides GMPLS control plane configuration, connection management and visualization of network topology, link state and neighbor discovery information, and light path connections.

Note that only the signaling, routing, and neighbor discovery and link management protocols are under standardization process. The design and implementation of these software modules and application modules are up to the vendors.

3.1 GMPLS application component

An application software component, called *GMPLS manager*, has been developed to act as the central point of distributed control plane. It coordinates the functions of all the GMPLS modules such as signaling, routing, and neighbor discovery and link management. It also provides the interface between the control plane and the actual data plane, i.e., the optical switch control point, allowing, for example, the set-up and tear-down of cross-connects, failure detection, as well as protection switching and reversion in the case of network failures. In addition, it can communicate with the external GUI/CLI terminal or EMS/NMS through the GUI/CLI agent for control plane configuration, topology display, connection management and display.

During initialization and boot-up, the GMPLS manager creates and initializes all other modules. It creates and deletes signaling and routing interfaces, and initializes related data structure in the signaling and routing components. Connection setup requests are received either from the management application or clients attached to the optical switch through an optical User-Network Interface[11]. In response to connection requests the GMPLS manager at the ingress node performs admission control, invokes the path computation module to compute an explicit route, and instructs the signaling component to initialize the path set-up along the explicit route. Path computation is subject to certain constraints, such as available bandwidth, diversity, and etc. Our architecture also allows for the explicit path to be provided by the management application, thus eliminating the need for path computation at the ingress node. Upon receiving connection teardown request from GUI/CLI or a client, the GMPLS manager instructs the signaling component to tear down the path.

At intermediate and egress nodes, the GMPLS manager handles LSP set-up/tear-down requests, which are passed up by the signaling component, and performs admission control and resource allocation. In the case of protected light paths, the GMPLS manager provisions the path based on the protection requirement (e.g. dedicated or shared mesh restoration) and performs restoration, which requires the computation of primary and backup paths, provisioning, switchover and reversion.

The GMPLS manager uses a protocol state machine to manage LSP operations. This approach facilitates multitasking implementation and error handling. It maintains a copy of information regarding the state of existing connections and tracks allocated and ready-to-use resources such as ports and labels. This speeds up the LSP provisioning and restoration process. The modular design of the GMPLS manager allows it to support multiple signaling protocols simultaneously, for example, CR-LDP and RSVP-TE. An additional important task of the GMPLS manager is to maintain consistency of all operations and roll back operations that might be incomplete.

The GMPLS manager consists of two sub-modules, the *GMPLS controller* that performs control functions, and *GMPLS adapter* that provides an interface to the routing and signaling components and shields from the details of underlining signaling, routing and link management component implementation.

3.2 Routing management

Information about the network topology and various attributes associated with links in networks is required to compute the path by the ingress node. Routing protocols are used to distribute the network topology and resource availability information. Traditional IP link-state routing protocols such as OSPF[12] and IS-IS[13] distribute only information about the state (up/down) of individual links and the administrative metric assigned to each link. GMPLS has extended IP routing protocols in order to support traffic engineering over different types of links, including optical links.

In general, there may be multiple links between a pair of nodes. This is particularly the case for core optical switches, which typically have hundreds of ports. Links between the same pair of nodes, with similar characteristics, can be bundled together and advertised as a single link bundle or TE link into the routing protocol for the purpose of traffic engineering in the GMPLS[14]. This reduces the amount of information handled by the routing components and improves routing scalability. At the same time, bundling leads to loss of information because it results in aggregation and information abstraction. It is, therefore, desirable to reduce the amount advertised by the routing protocol without losing too much necessary information. In accordance with the proposed GMPLS standards, a link bundling mechanism has been developed, which automatically classifies the component links into unnumbered link bundles based on their cost, rate, and SRLGs. Component links are potentially reclassified into another link bundle after their characteristics change, for example, when some channels on a link have become used or the maximum reservable bandwidth of the link has changed.

The Routing Management component is responsible for managing the link state database and routing tables by supporting the routing protocols. It uses standard routing protocols to flood the network routing information. It also provides the interface for passing the routing information to a routing component user, allowing it to access the routing information, make routing decisions, and obtain an explicit route for a connection. Furthermore, it can also provide asynchronous notification to registered users about changes in the routing information. The Routing Management component can be sub-divided into several modules, including *routing engine*, *routing table manager*, and *path computation*.

The routing engine sub-module provides the implementation of standard routing protocols. It updates the link state database and distributes the routing information about network topology and optical link attributes using routing protocols. Our implementation uses the OSPF protocol with appropriate traffic engineering and GMPLS extensions. It is assumed that there is a single OSPF adjacency between two neighboring nodes. For the purpose of routing scalability as described above, multiple physical optical links between two adjacent nodes with the same characteristics are aggregated into an unnumbered link bundle when advertised by OSPF. In addition to Router Link State Advertisements (LSAs), traffic engineering LSAs are also advertised by the extended OSPF routing engine. Resource attributes associated with optical link bundles are mapped into the data structure of extended OSPF traffic engineering LSA so that information about the state of resources in the optical network is disseminated. The LSA database contains self-originated LSAs and LSAs received from other nodes. It is populated and updated by the routing engine. The routing engine is bound to routing table manager on the upper edge and the layer 3 stack (e.g. IPV4) on the lower edge.

The path computation sub-module is responsible for computing the path for the requested connections within the optical network. It uses the routing information in the link state database by taking into account a set of constraints such as protection level, bandwidth, and traffic engineering requirements. Note that path computation algorithm is not subject to standardization since it allows vendors to innovate on algorithms that may provide differentiated performance. However, what is the subject of standardization is the information exchanged between different equipment in order to feed into the algorithms performing the path computation. A constraint-based K-shortest path computation algorithm is employed for our path computation. It supports various path protection schemes, such as unprotected, 1+1 path protection, and shared mesh restoration. In the case of protection, the primary and backup paths can be SRLG disjoint or SRLG and node disjoint. The path computation is invoked through the routing table manager by the GMPLS manager at the ingress node when a LSP is set up. The parameters and constrains for the requested optical path or paths (in the case of protection) such as source and destination nodes, bandwidth, diversity, protection type, and etc. are passed to the path computation module as input. The output of the path computation is in the form of a list of nodes and the link bundle on each node that consist of the primary and backup (in the case of protection) paths.

The final sub-module within the Routing Management Component is the routing table manager (RTM), which provides the interface between the GMPLS manager and the routing component. The GMPLS manager and other users can access the routing information and obtain explicit route through the RTM. Thus, the RTM isolates external modules from the specifics of the implementation of the routing engine and path computation. It allows the abstraction of any routing protocol that can be used transparently by other GMPLS modules. This simplifies the integration of routing implementation. The RTM can support multiple routing protocols (e.g. OSPF and BGP[15]) and path computation algorithms simultaneously. OSPF-TE is currently implemented as the only routing protocol. The GMPLS manager also updates the routing engine through the RTM about the changes on the local link configuration (e.g. an LSP is set up and, as a result, the available bandwidth on this link is reduced). The routing engine through OSPF floods changes in the link state database. A threshold policy with configurable parameters can be implemented for advertising changes of different magnitude.

3.3 Signaling component

Signaling is needed to establish the LSP path. GMPLS extends the two signaling protocols (RSVP-TE[7] and CR-LDP[16]) to support generalized interfaces. It is the manufacturer's choice which protocol is used. Generalized RSVP-TE is employed as the signaling protocol in our implementation, which is the dominant signaling protocol. However the software architecture does not preclude other protocols. In fact, multiple signaling protocols can be supported simultaneously by the GMPLS manager through the interfaces provided by the GMPLS adapter. The GMPLS control plane currently supports provisioning unprotected, 1+1 protected, and shared mesh restored light paths based on the RSVP-TE protocol with GMPLS extensions.

The RSVP-TE signaling component is responsible for establishing and closing signaling sessions with neighbor signaling peers. It processes LSP-related requests from the GMPLS manager via the interfaces provided by the GMPLS adapter, generating appropriate control messages, and sending them along the path for label request and resource reservation/allocation during a new LSP setup or for label and resource release during a LSP deletion. It receives and decodes incoming signaling messages and sends the appropriate notifications to the GMPLS manager (for examples, label-related notifications and traffic control-related notifications). In our implementation, the signaling component sends these notifications to the GMPLS manager using application callback functions through the interfaces provided by the GMPLS adapter. During the initialization, the signaling component is bound to the GMPLS manager on the upper edge via interfaces provided by GMPLS adapter and to the Layer 3 stack (IPV4) on the lower edge for sending and receiving messages.

The path computed by the ingress node is placed in the Explicit Route Object (ERO), carried in the RSVP PATH message. The Explicit Route Object (ERO) contains the list of all the nodes and corresponding link bundles used on each node for the primary path and 1+1 backup path. In the case of the shared mesh restored backup path, only node information is carried in the ERO in order to maximize sharability. Strict ERO is employed and will not be modified by intermediate nodes along the path. Note that the ERO only determines the node/link bundle to be used for the path. Selection of specific component link in the bundle between two adjacent nodes is made by the upstream node according to its local policy. For the shared mesh restoration, the SRLG/node information of the primary path is also carried in the signaling message in order to determine the sharable channels.

3.4 Neighbor discovery and link management

There may be multiple port/fiber links between two neighbor nodes. These links may be bundled into a single TE link bundle for routing purposes. Furthermore, the control channels for routing, signaling, and link management may not use the same physical medium and interfaces as the data links. Neighbor discovery and link management provides the capability to manage control and data links between neighboring nodes. It can establish and maintain control channel connectivity, discover and correlate the physical properties of data links, verify the data link connectivity, localize and handle link failures. The control channel between two adjacent nodes can be established and maintained through configuration message exchange and keep-alive mechanisms. The data link connectivity and interface ID can be discovered and verified through in-band test messages. The information about configuration and properties of data links can be discovered and correlated through exchange of link summary messages between two neighboring nodes, which aggregate multiple data links into a TE link bundle and synchronize the properties of the TE link bundle. In addition, the link fault management scheme can be used between adjacent nodes to localize failures so that local span or end-to-end path protection and restoration procedures can be initiated.

3.5 GMPLS database

Information about the link state and path connections are needed to compute, provision and maintain the optical paths, and provide network survivability using protection and restoration techniques in case of network failure. A database is maintained in each node, which stores the necessary information. It contains several tables. The port table stores information about the characteristics and states of ports in the optical switch, including port type, port data rate, remote node and port ID, resource class, and operational and administrative status. The link bundle table includes the information about link bundle properties, the list of its component links, local and remote link bundle IDs. The port and link bundle database information can be populated through manual configuration or through link management mechanism that automatically discover neighbor port characteristics and state. Path table stores connection-related information such as path ID, path name, path type, associated path ID (for protection and restoration), ingress and egress ports, labels, and etc. The database manager is developed to provide an interface for the GMPLS manager to access the database. It shields the GMPLS manager from the database implementation details and location (the databases may reside in the memory of the same control card or on the different card or disk). The GMPLS manager uses the database for path setup, admission control, port, channel and label assignment, and etc.

3.6 GUI/CLI agent

GUI/CLI agent is not a mandatory component of the GMPLS software suite. However it provides an interface to external host or EMS/NMS system for configuration and information display. For example, GMPLS protocol configuration such as timer values for RSVP refresh interval and number of retries in the case of path setup failure are initialized to default values at bootup, but may be changed by users through GUI/CLI agent. Certain database information can be populated. In addition, the LSP manual setup and teardown commands can be sent to the ingress node from the management station. The network information such as LSP connection, protection and restoration can also be sent to the management station for display using GUI/CLI agent. The GUI and CLI provide a useful tool for GMPLS software testing and debug, and network operation and protocol verification.

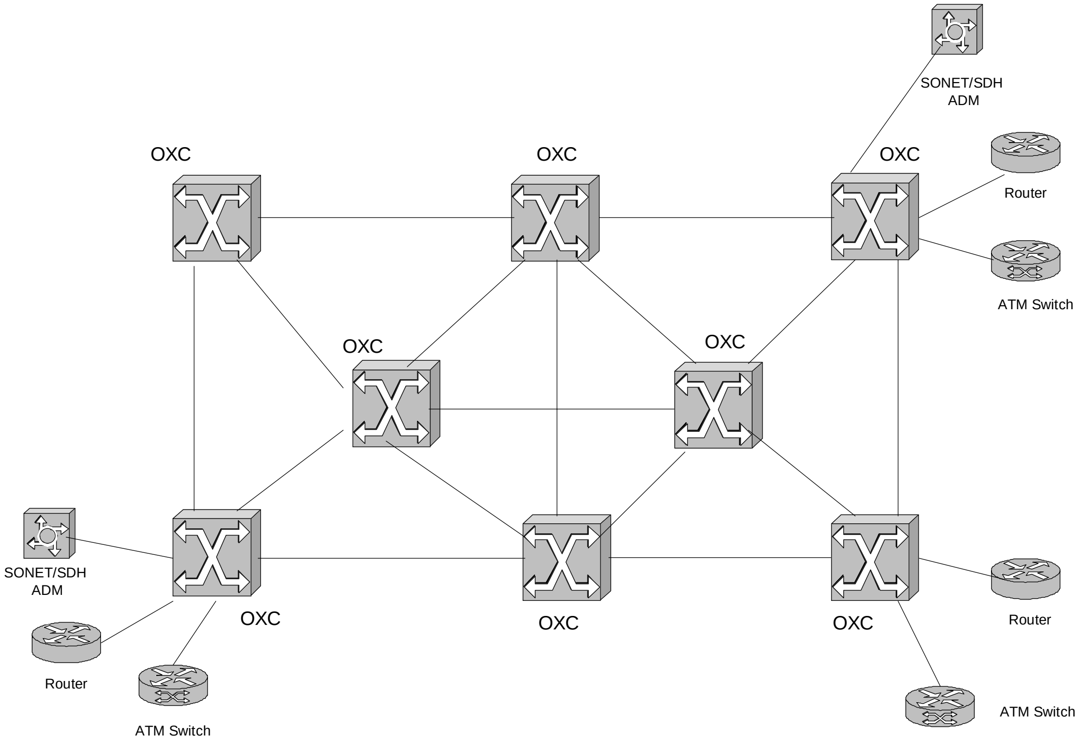

Figure 2: optical network architecture.

4. EXPERIMENTAL NETWORK

Figure 2 depicts a typical network architecture with the optical switches in the core network and other equipment such as routers and ATM switches connected to the network as client nodes. The behavior of a client node is not exactly the same as the behavior of a network node. The interface between a client and a network node (referred to as a User to Network Interface - UNI) has some special characteristics compared to the interface between two network nodes. Some of these characteristics have not been addressed by the GMPLS. However, GMPLS has been enhanced to support the UNI characteristics by other standardization bodies like the Optical Internetworking Forum (OIF). The OIF defines an interface between client equipment and a network on the top of GMPLS to support UNI characteristics[11]. With OIF UNI interface, a client can send an LSP setup request along with destination client address and required LSP parameters such as bandwidth, the desired protection and diversity, etc. to one of connected network node. The network node computes the LSP according to the network routing information, the requested LSP parameters and traffic engineering requirements and establishes the LSP in the networks.

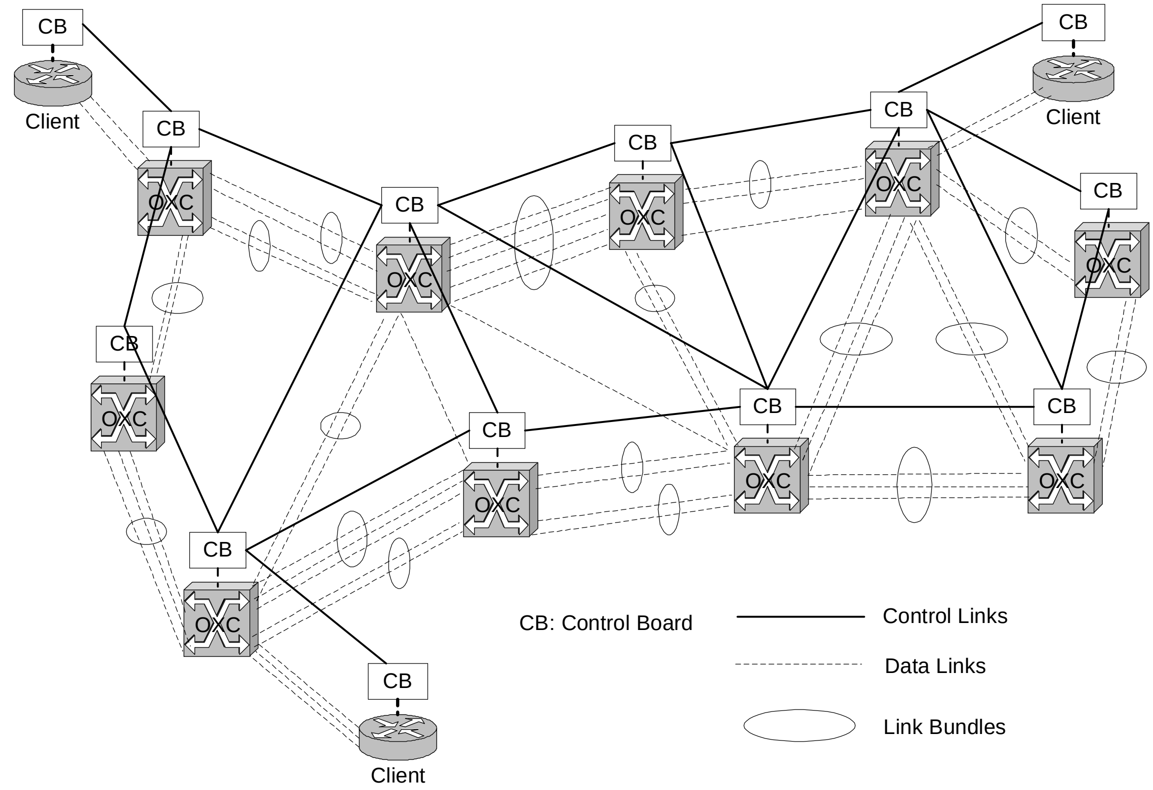

Figure 3: GMPLS prototype network.

In order to study GMPLS control plane for provisioning the optical LSP and providing protection and restoration among the nodes in the core networks, a prototype network has been developed. Figure 3 shows a topology of prototype network, which consists of 10 network nodes and 3 client nodes. Larger networks with more network and client nodes were also tested. In the prototype networks, the links with the same end nodes and characteristics (rate, type, SRLGs, etc.) are grouped into a TE link bundle, which allows summarization and reduction of information distributed by OSPF-TE. RSVP-TE with GMPLS extensions is used as the signaling protocol in the core network and the OIF UNI 1.0 is used to invoke the network services by the client nodes.

A host computer with customized GUI-based management software is connected to the experimental network to simulate the EMS/NMS. The GUI is used to provision the network and initiate light path set-up and deletion. The network topology and the path connections as well as their parameters are also displayed on the GUI.

With the GMPLS prototype network, we have demonstrated the network routing information distribution through extended OSPF, dynamic light path computation and provisioning with 1+1 protection and shared mesh restoration using extended RSVP-TE and OIF UNI signaling.

5. CONCLUSIONS

This paper presented the design considerations and the architecture of a GMPLS-based control plane software targeted for core optical networks deployed in mesh topology. GMPLS is intended to provide a scalable, interoperable and distributed network control plane for multiple types of network equipment. The prototype has demonstrated the feasibility, main features and benefits of GMPLS as a common control plane to support distributed topology dissemination, dynamic path computation and provisioning, protection and restoration in the optical core network. Various requirements (traffic engineering, desired protection and bandwidth requirements) are taken into consideration during the LSP establishment to optimize the network utilization. Our initial results lead to the conclusion that the general concept of a GMPLS-based control plane is feasible with careful architecture and design of the software components. Of course, much further work remains such as detailed performance evaluation of the operation of GMPLS in a fairly large size network consisting of various types of equipment. In addition to such feasibility results, actual adoption and deployment of GMPLS as a common control plane for intelligent transport networks will depend on the successful completion of relevant standardization activities, extensive interoperability testing as well as the strengthening of appropriate business drivers.

ACKNOWLEDGEMENTS

The authors would like to thank Sid Chaudhuri and Bala Rajagopalon for useful input that they have provided in this work.

REFERENCES

1. L. Andersson, et al., “LDP Specification,” *RFC 3036*, January 2001.
2. B. Jamoussi, et al., "Constraint-Based LSP Setup using LDP," *RFC 3212*, January 2002.
3. R. Braden, et al., "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification," *RFC 2205*, September 1997.
4. E. Mannie, Editor, “Generalized Multi-Protocol Label Switching (GMPLS) Architecture,” *Internet Draft*, work in progress, draft-ietf-ccamp-gmpls-architecture-02.txt, March 2002.
5. P. Ashwood-Smith, et al, "Generalized MPLS - Signaling Functional Description," *Internet Draft*, work in progress, draft-ietf-mpls-generalized-signaling-08.txt, April 2002.
6. D.O. Awduche, et al., "RSVP-TE: Extensions to RSVP for LSP Tunnels,” *RFC 3209*, December 2001.
7. P. Ashwood-Smith, et al., "Generalized MPLS – RSVP-TE Extensions," *Internet Draft*, work in progress, draft-ietf-mpls-generalized-rsvp-te-07.txt, April 2002.
8. D. Katz, et al., "Traffic Engineering Extensions to OSPF", *Internet Draft*, work in progress, draft-katz-yeung-ospf-traffic-06.txt, October 2001.
9. K. Kompella, et al., “Routing Extensions in Support of Generalized MPLS,” *Internet Draft*, work in progress, draft-ietf-ccamp-gmpls-routing-04.txt, April 2002.
10. K. Kompella, et al., "OSPF Extensions in Support of Generalized MPLS", *Internet Draft*, work in progress, draft-ietf-ccamp-ospf-gmpls-extensions-05.txt, April 2002.
11. Bala Rajagopalon, Editor, “User Network Interface (UNI) 1.0 Signaling Specification,” *Optical Internetworking Forum*, October 2001.
12. J. Moy, "OSPF Version 2", *RFC 2328*, April 1998.
13. D. Oran, " OSI IS-IS Intra-domain Routing Protocol,” *RFC 1142*, February 1990.

14. K. Kompella, et al., "Link Bundling in MPLS Traffic Engineering," *Internet Draft*, work in progress, draft-kompella-mpls-bundle-05.txt, February 2001.
15. Y. Rekhter and T. Li, “A Border Gateway Protocol 4 (BGP-4),” *RFC 1771*, March 1995.
16. P. Ashwood-Smith, et al., "Generalized MPLS Signaling - CR-LDP Extensions," *Internet Draft*, work in progress, draft-ietf-mpls-generalized-cr-ldp-06.txt, April 2002.
17. J.P. Lang, et al., "Link Management Protocol (LMP)," *Internet Draft*, work in progress, draft-ietf-ccamp-lmp-03.txt, March 2002.