A GMPLS optical control plane for IP/Gigabit Ethernet over dynamic DWDM networks. Testbed implementation for soft-permanent connections

R. Muñoz, C. Pinart, G. Junyent

Publication:	7th IFIP Working Conference on Optical Network Design and Modeling (ONDM 2003)
Vol.:	2
No.:	-
pp.:	801-820
Date:	Budapest (Hungary). February 3-5, 2003

A GMPLS optical control plane for IP/Gigabit Ethernet over dynamic DWDM networks

Testbed implementation for soft-permanent connections

Raül Muñoz[1], Carolina Pinart[1] and Gabriel Junyent[2]

1: Centre Tecnològic de Telecomunicacions de Catalunya (CTTC), c/Gran Capità 2-4, 08034 Barcelona, Spain, {raul.munoz, carolina.pinart}@cttc.es, 2: Universitat Politècnica de Catalunya (UPC), c/Jordi Girona 1-3, 08034 Barcelona, Spain, junyent@tsc.upc.es

Abstract: This paper presents an experimental study of the ASON/GMPLS optical control plane that is being implemented on the testbed developed at CTTC laboratories, focusing on the setup and teardown of optical connections for Ethernet traffic triggered by the Network Management System, and considering not only architectural, conceptual and functional requirements that can be found in the existing literature but also, and especially, on design and implementation issues. Hence, we give insight into the design and implementation of the optical control plane, focusing on the modules, protocols and mechanisms, such as topology and optical resources discovery, route computation and wavelength allocation, necessary for ligthpath setup and teardown Is out of the scope of this paper to give performance results of the proposed architecture.

Key words: GMPLS, ASON, SPC, RWA, CONTOL PLANE, DWDM, ETHERNET

1. INTRODUCTION

The rapid growth of data traffic is undeniable thanks to Internet explosion, showing clearly that the existing IP/SDH/ATM/WDM infrastructure (Figure 1a), multibillion-euro expensive to deploy, is incapable of handling efficiently such a data traffic load, for it has static bandwidth availability (i.e. weeks time to get a new link deployed) and bandwidth demand in huge increments. This situation is motivating the research community to look to ever higher-speed network technologies capable of reducing deployment costs, of solving the bandwidth demand crunch and of providing flexible and dynamic bandwidth (just-in-time provisioning). In this context, extending Ethernet LAN to flexible, dynamic DWDM-switching all-optical networks for MAN and WAN is a very interesting solution.

The proven fact that most of the data traffic is IP/Ethernet, along with the fact that Ethernet technology offers the possibility to provide the demanded capabilities for next-generation networks, makes Ethernet an attractive backbone technology, capable of overthrowing the current SDH/ATM infrastructure. However, migrating today's networks to an all-optical framework using Ethernet as backbone technology is complex, since many questions abound, such as the reliability/survivability or the guarantees of QoS. In this context, the Automatic Switched Optical Network (ASON) [1] is emerging as a promising candidate to solve many of Ethernet's lacks, due to its capability of providing flexible, dynamic, end-to-end optical connections as well as optical protection and restoration, among other functionalities. In ASON, these functionalities are supported by an optical control plane that is responsible for introducing intelligence on the optical transport network. Although there is an important effort to standardize the architecture and functional requirements of ASON networks, fundamentally of the optical control plane, a considerable effort is still required for the practical deployment of this type of network, especially IP/Ethernet/ASON (Figure 1b). In this context, the IETF has proposed *Generalized Multi-Protocol Label Switching* (GMPLS) [2] as a control plane for ASON networks by means of IP routing and signalling protocols, allowing integration with classical IP networks.

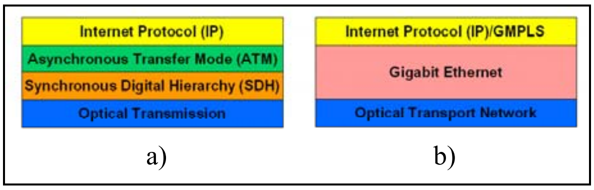


Figure 1. a) Layering of current SDH/ATM b) Layering of Ethernet-based architecture

In this paper we present the optical control plane implementation of the ASON/GMPLS testbed for Ethernet traffic being developed at CTTC laboratories for dynamic setup and teardown of optical connections triggered by the Network Management System (NMS). This paper is organized as follows. In Section 2 we present the network model of the ASON/GMPLS testbed and the types of lightpath triggers. Section 3 describes the architecture of the implemented optical control plane, detailing the implemented modules and their functionalities, such as neighbour discovery, topology discovery, available optical resources, route computation, wavelength assignment or lightpath setup. Finally, Section 4 is devoted to the conclusions of this paper and presents briefly future work on the testbed.

2. NETWORK MODEL OVERVIEW

Our testbed is based on an ASON/GMPLS network constituted by a metropolitan wavelength division multiplex ring using dense channel spacing (DWDM) with three dynamically configurable all-optical add-drop multiplexers (OADM), allowing, among other functionalities, the establishment of real-time, dynamic, end-to-end connections between Gigabit Ethernet client equipment. Figure 2 shows the logical architecture of our testbed, which can be characterized by three planes: transport, control and management.

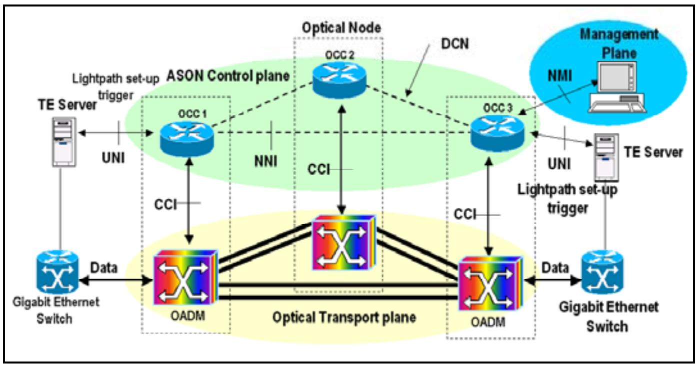


Figure 2. Logical architecture of the implemented ASON/GMPLS testbed

The transport plane is the optical transport network (OTN), and is responsible for the provision of uni- and bi-directional optical channels transparent to the format and payload of client signals, and for the detection of information about the state of connections, such as errors or signal quality. The control plane supports setup and teardown of dynamic, real-time optical connections by means of routing and signalling protocols, using the Network Node Interface (NNI) to communicate with other nodes, the User Network Interface (UNI) to communicate with Traffic Engineering Servers (TES), and the Connection Control Interface (CCI) to configure optical node's hardware. The management plane performs operations and management functions, such as failures, network elements configuration, security or billing, using the Network Management Interface (NMI) to communicate with the control and transport planes. A more detailed description of the transport and management planes can be found in [1]. Hence, and since they are out of the scope of this paper, their architecture will not be discussed any further. Instead, the remainder of this paper will concentrate on the control plane.

Our testbed considers two types of triggers for the optical connection establishment: through the network management system (NMS) or by means of TES. In the first case, the NMS requests the establishment of an optical connection to the source optical node using the NMI interface based on *Simple Network Management Protocol* (SNMP). The source node uses the routing and signalling protocols of the control plane to set up the optical connection to the destination node using the NNI interface. Finally, the source node reports the NMS about the success or failure of the connection requested. This type of connections are named *Soft-Permanent* (SPC) and are the focus of this paper. SPCs are considered hybrid due to the fact that both the control and management planes are involved. However, SPCs do not take full profit of control plane functionalities, since they are triggered by a network operator, that is, they are not triggered automatically. Therefore, once the NMS has received the acknowledgement of an optical connection, the client equipment configuration depends on the network operator.

In the second case, each edge optical node has a TES that monitors and collects information via an SNMP interface, such as traffic volume in each port, lost packets or available resources about client equipment (Gigabit Ethernet routing switches) connected to the optical nodes. Such information is used by the TES to estimate unfavourable situations in which the client equipment could need more optical connections. Once a TES has detected a critical situation, it requests the control plane a new optical connection using the UNI interface, standardized by the OIF [3]. In case of successful reservation of the required resources, the control plane reports to source and destination TES, which configure dynamically the client equipments involved. This connection type, called switched, is out of the scope of this paper, and therefore no traffic engineering function is implemented

3. OPTICAL CONTROL PLANE ARCHITECTURE

The optical control plane of an ASON network has the main function of providing the optical transport network with intelligence, allowing, by means of routing and signalling protocols, dynamic, flexible, real-time provisioning of optical channels, traffic engineering for allocation of routes and resources (wavelengths), protection/restoration, QoS and optical VPNs.

The optical control plane can be based on IP or ATM routing and signalling protocols. Initially, the IETF proposed to adapt IP-based protocols for the control plane, particularly on the MPλS control plane, that is essentially the *Multi-Protocol Label Switching* (MPLS) control plane with extensions for wavelength switching. More recently, *Generalized MPLS* (GMPLS) has also been proposed, which extends MPLS to support multiple types of switching such as packets, time division multiplex (SDH/SONET time slots), wavelengths and fibre optics. In the last months, it has also been proposed to base the optical control plane on ATM protocols, such as the Private NNI signalling protocol (PNNI). In our testbed we have adopted the GMPLS control plane to support lambda switching. The control plane we have implemented (Figure 3) is based upon a *Data Communication Network* (DCN) and three *Optical Connection Controllers* (OCC), detailed in Sections 3.1 and 3.2.

3.1 Data Communication Network

The DCN provides data channels with IP connectivity, allowing the exchange of control messages between neighbour OCCs and of management messages between the NMS and the OCCs. The DCN has been implemented trough full duplex Fast Ethernet links transported out-of-band on wavelengths of 1310 nm. There is not a one-to-one association of a control channel to a fibre, that is, there is only a single control channel for all the fibre between to neighbour OCCs. Each OCC has been implemented on a Linux platform with two 1 GHz processors, acting like an IPv4 router.

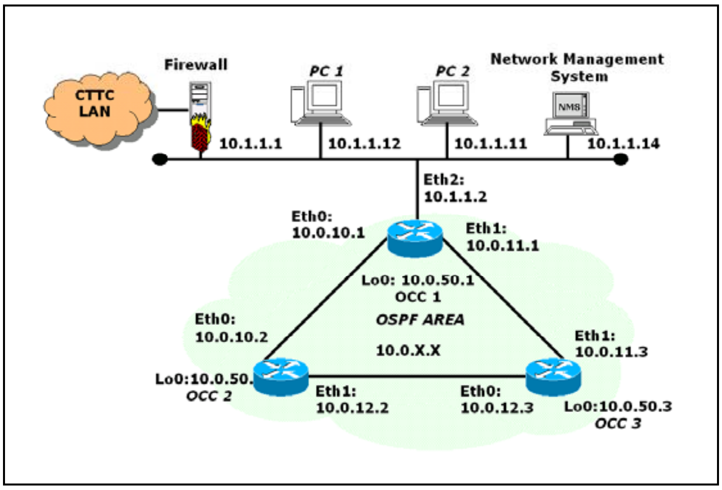


Figure 3. Data Communication Network

3.2 Optical Connection Controller

OCCs are responsible for handling OTN's resources in order to manage and monitor the establishment and deletion of optical connections through the exchange of control messages. Each OCC has been designed following an architecture similar to the one proposed in ITU-T Recommendation G.8080 [1], not considering Traffic Policing, Call Control, authentication nor encryption. Note that the architecture proposed in [1] only describes the control plane in terms of modules that represent abstract entities, defining the provided services of each controller, as well as its interfaces, without specifying which routing or signalling protocols must be used or implementation details. Instead, our focus is to provide implementation and design details of each controller for SPC. As depicted in Figure 4, the implemented modules are the *Connection Controller* (CC), which manages optical connection establishment and deletion and coordinates the rest of modules, *Routing Controller* (RC), responsible for network topology discovery and route computation, *Link Resource Manager* (LRM), responsible for local optical resources (wavelengths and fibres) management and wavelength allocation, and *Protocol Controller* (PC), which provides the function of mapping the parameters of the controller's interfaces into messages that are carried by a protocol to support interconnection via an interface. We have implemented four PC: lightpath signalling and routing exchange (NNI), management (NMI) and optical hardware configuration (CCI).

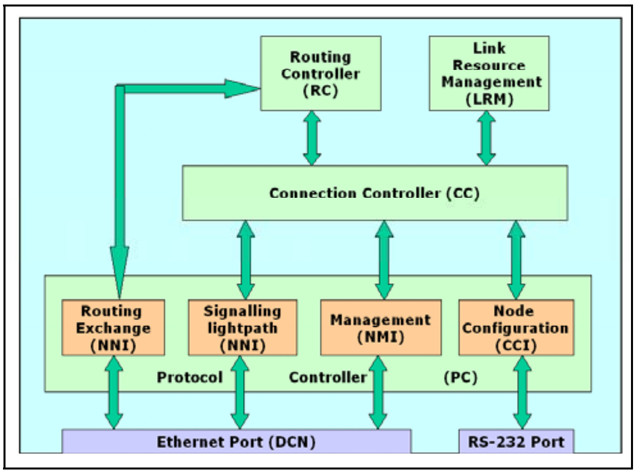


Figure 4. OCC architecture

3.3 Routing Controller and Link Resource Manager

The main function of these controllers is to collect on databases information about topology and available resources of the optical network to be used by the CC. When an OCC receives a request to set up an optical connection, the problem of assigning a path over which the connection must be established (routing problem) and of assigning a wavelength on each link along the selected path (wavelength assignment problem) is known as Routing and Wavelength Assignment (RWA) [4]. There are three variants of the RWA problem, depending on the type of optical connection requests: static, incremental or dynamic. Our testbed is based on dynamic requests triggered by the NMS, in which a lightpath is set up for each connection request as it arrives, and the lightpath is released after a finite amount of time. The objective is to set up lightpaths and to assign wavelengths minimizing the connection blocking probability.

3.3.1 Routing Controller: routing problem, network topology discovery and route computation

Relating to the routing problem, it has three possible solutions: *Fixed Routing*, *Fixed-Alternate Routing,* and *Adaptive Routing*. In *Adaptive Routing*, the path between a source and a destination is calculated dynamically depending on the network state and the likelihood of success on establishing connections depends on the network state information availability. Moreover, *Adaptive Routing* can be implemented in either a centralized or distributed manner. Our testbed considers distributed adaptive routing based on global information, allowing OCCs to have full information about the optical network topology, but not about the available resources on each link. A distributed adaptive routing based on global information can be implemented in a link-state approach (OSPF, IS-IS) or in a distance-vector approach (RIP). Our testbed is based on the OSPF link-state routing protocol to distribute the network topology information (Figure 5).

The optical network topology can be viewed as a uni-layer graph in which each link in the network is associated with a specific weight function that denotes the cost of using the link. Each link of the graph between to nodes represents all the physical links (fibres) between these two nodes. A way of selecting a path from a source to a destination is to determine the shortest-cost path using Dijkstra's algorithm, in which the weight functions depend on several factors, such as the hops (lightpaths are selected solely on the smallest number of hops on each lightpath) or distance (weight functions represent physical distance). [5] proposes to incorporate DWDM-specific information in the weight functions to improve the performance of Dijkstra’s algorithm, through considering the number of available wavelengths and total wavelengths on a given link.

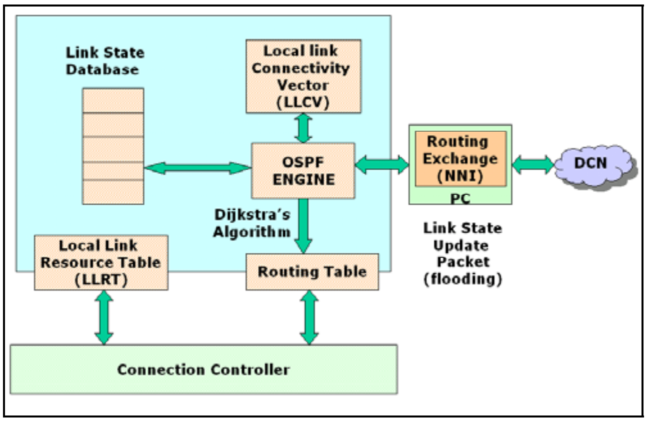


Figure 5. LRM ad RC architecture

Figure 6a shows the weighted graph representing the network topology of our testbed at a given time. Each link of the graph is associated with a weight function based on the number of available wavelengths, so that when the number of available wavelengths on a link is 0, the cost of the link becomes ∞. The weighted graph can be modelled by an adjacency matrix that we have named *Global Link Connectivity Table* (GLCT) (Figure 6b), which is used by Dijkstra’s algorithm to calculate the shortest-cost path and to generate the *Routing Table*. Each k*th* row corresponds to the local connectivity of the k*th* OCC, which we have named *Local Link Connectivity Vector* (LLCV), and each j*th* element of the LLCV represents the link cost with the j*th* OCC.

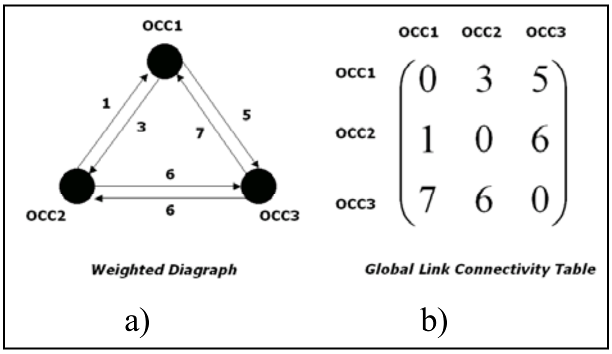


Figure 6. a) Weighted graph of the network topology and b) adjacency matrix (GLCT)

The weight functions of the LLCV are configured automatically whenever there is a change on the available wavelengths on each node. So, each OCC distributes its local state (LLCV) throughout the optical network by flooding *Link State Update* messages and collects all the received LLCV on a database containing the network topology (*Link State Database*, Figure 5). Each OCC has an identical *Link State Database*, in which each individual piece is a particular OCC's LLCV. Hence, the *Link State Database* describes the GLCT. Note that no constraint shortest path algorithm is used to implement Traffic Engineering functions.

3.3.2 Link Resource Management: wavelength assignment problem and local optical resources

The wavelength assignment problem appears when there are multiple wavelengths from a source node to a destination node, and consequently a selection algorithm is needed. This algorithm is based on heuristic methods that try to minimize connection blocking probability. In the literature, several classic heuristic algorithms can be found, such as First-Fit, Least-Used, Most-Used, Min-Product, Least Loaded, Max-Sum, Relative Capacity Loss, Wavelength Reservation or Protection Threshold [4], which can be divided into those requiring global information about the optical resource availability (fibres and wavelengths) and those only requiring local information. Among the before-mentioned algorithms, only the Random and First-Fit do not require global knowledge of resources availability. On our testbed, as described previously, the available information about optical resources is local, and therefore each OCC has a LLRT containing all the information about fibres and wavelengths locally available. Hence, we consider the First-Fit algorithm due to its good performance, small computational overhead and low complexity.

First-fit is based on packing all of the in-use wavelengths towards the lower end of the wavelength space so that continuous longer paths towards the higher end of the wavelength space will have a higher probability of being available. In order to collect all the information about available resources in the LLRT of each OCC, an addressing scheme for all the optical resources of the network must be defined. The simplest option consists of assigning an IP address per wavelength, but, since this option clearly represents a great waste, the IETF proposes the use of *Unnumbered Links* and *Bundled Link* addressing schemes. The *Unnumbered Links* addressing scheme consists of providing each OCC with only an IPv4 address acting as node identifier (NODE_ID). The concept of logical port is used here to identify each wavelength, formed by the physical port identifier (PORT_ID), which identifies the different fibres of the optical node, and the channel identifier (CH_ID), which identifies the different wavelengths inside an optical fibre. The *Bundled Link* addressing scheme allows to group different links that share common characteristics under the same identifier, with the aim of reducing the computational overhead. Figure 7 shows an example of the addressing scheme for OCC1 in our testbed, based on both *Unnumbered Links* and *Bundled Link*.

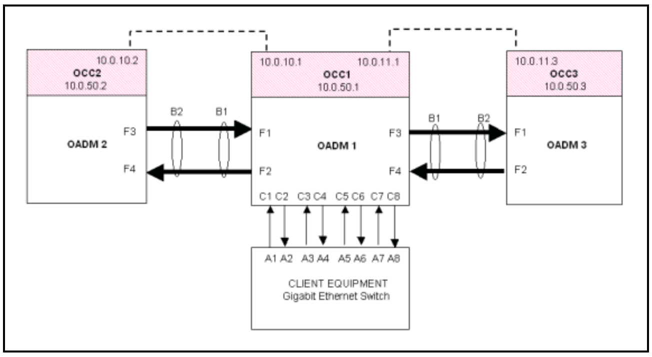
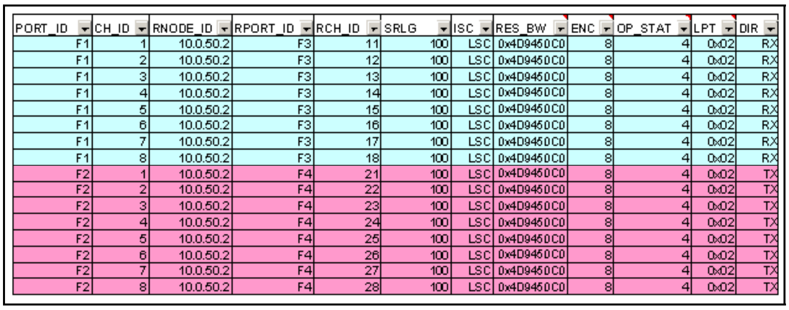


Figure 7. OCC1 addressing scheme

Note that the above-mentioned identifiers only have value locally, that is, each OCC may assign different identifiers to the same physical fibre or wavelength. Therefore, each OCC must register its local optical resources in the LLRT, and map its identifiers with the identifiers of its neighbour OCCs (remote identifiers). On our testbed, the LLRT is manually configured, although we also consider future automatic configuration by means of an automatic neighbourhood mechanism such as the LMP, which would allow automatic detection of new fibres and wavelengths between an OCC and its neighbours, and to update LLRTs with local and remote identifiers. In the same way, this mechanism would allow to detect optical resource failures and to update the corresponding LLRT.

Figure 8 illustrates an example of the LLRT for OCC1 on our testbed, in which each wavelength is identified by its PORT_ID (local fibre identifier), CH_ID (local wavelength identifier), RNODE_ID (local node identifier), RCH_ID (remote wavelength identifier), SRLG (administrative group associated to specific optical resources that share vulnerability in the event of a failure), ISC (switching type supported: packet, slot, lambda or fibre, taking the value of *Lambda Switch Capable*, LSC, on our testbed), RES_BW (maximum bandwidth supported by the optical channel), ENC (type of the client signal transported), OP_STAT (resource state: *Unequipped, Reserved, In-Service, Failed or Stand-by*), LPT (link protection type: *Unprotected, Extra Traffic, Shared, Dedicated 1+1, Dedicated 1:1* or *Enhanced*) and DIR (transmission or reception). Note that the reserved status indicates that the wavelength has been reserved for a requested connection. The signalling scheme is based upon downstream reservation, in which wavelength resources are reserved along the downstream path to the destination on a hop-by-hop basis. Although this status may also indicate that the wavelength has been reserved for protection of another optical channel, it is out of the scope of this paper, since no type of protection based on the control plane is contemplated.



PORT_ID	CH_ID	RNODE_ID	RPORT_ID	RCH_ID	SRLG	ISC	RES_BW	ENC	OP_STAT	LPT	DIR
F1	1	10.0.50.2	F3	11	100	LSC	0x4D9450C0	8	4	0x02	RX
F1	2	10.0.50.2	F3	12	100	LSC	0x4D9450C0	8	4	0x02	RX
F1	3	10.0.50.2	F3	13	100	LSC	0x4D9450C0	8	4	0x02	RX
F1	4	10.0.50.2	F3	14	100	LSC	0x4D9450C0	8	4	0x02	RX
F1	5	10.0.50.2	F3	15	100	LSC	0x4D9450C0	8	4	0x02	RX
F1	6	10.0.50.2	F3	16	100	LSC	0x4D9450C0	8	4	0x02	RX
F1	7	10.0.50.2	F3	17	100	LSC	0x4D9450C0	8	4	0x02	RX
F1	8	10.0.50.2	F3	18	100	LSC	0x4D9450C0	8	4	0x02	RX
F2	1	10.0.50.2	F4	21	100	LSC	0x4D9450C0	8	4	0x02	TX
F2	2	10.0.50.2	F4	22	100	LSC	0x4D9450C0	8	4	0x02	TX
F2	3	10.0.50.2	F4	23	100	LSC	0x4D9450C0	8	4	0x02	TX
F2	4	10.0.50.2	F4	24	100	LSC	0x4D9450C0	8	4	0x02	TX
F2	5	10.0.50.2	F4	25	100	LSC	0x4D9450C0	8	4	0x02	TX
F2	6	10.0.50.2	F4	26	100	LSC	0x4D9450C0	8	4	0x02	TX
F2	7	10.0.50.2	F4	27	100	LSC	0x4D9450C0	8	4	0x02	TX
F2	8	10.0.50.2	F4	28	100	LSC	0x4D9450C0	8	4	0x02	TX

Figure 8. Local Link Resource Table

Finally, we require the wavelength-continuity constraint due to the fact that our OTN is an OADM ring without wavelength conversion, and therefore a lightpath must occupy the same wavelength on each link in its route. To achieve the wavelength-continuity constraint, there are two possible solutions: the first one is based on the availability of global information about network resources, in which each OCC distributes its LLRT throughout the optical network by OSPF flooding, allowing to find the shortest cost path from a source node to a destination node with the wavelength-continuity constraint.The second is aggressive reservation. The solution implemented on our testbed is based on the aggressive reservation approach: when a new connection request arrives to the source OCC, it uses the RC to determine a strict path to the destination. It then chooses a set of available wavelength using the First-fit algorithm, and attempts to reserve at least one wavelength of this set along the entire path by propagating a control message to the next node with the Label Set object. Note that there is no guarantee that at least one wavelength of selected set be available along every link in the path; if the wavelength is blocked (no wavelength of the set is available in each link in the path), the source node can either choose another set of wavelengths or an acceptable label set sent by the node that has blocked the connection previously, and the process is repeated until there was at least one wavelength available. In case there is no wavelength available, the request is blocked and the lightpath cannot be set up. This approach has three major problems: (i) network resources are over-reserved for a short period of time, which might lead to the blocking of subsequent connection requests and to lower network utilization, (ii) it may result in long setup times, since it may take several attempts before a node can establish a lightpath, and (iii) there may be other paths from the source to the destination with wavelength continuity not selected by Dijkstra's algorithm as the shortest, and consequently the source node will not attempt to establish the lightpath on these paths even if the connection request is blocked. Despite these problems, we have chosen the second solution for our testbed because it allows having local information about optical resources. Note that the current GMPLS-RSVP signalling does not allow the implementation of the aggressive reservation approach for the upstream label of a bi-directional connection request, since the Upstream Label object only specifies a single downstream wavelength and there is no guarantee that this single wavelength chosen by the source node be available along every link in the path.

3.4 Connection Controller: GMPLS Daemon

The CC is the main controller of the OCC. Its function is to receive and process optical connection requests and packets from the TES, the NMS and the neighbour OCCs through the respective interfaces, managed by the PC. Figure 9 shows the implemented architecture of the CC and their associated PC for soft-permanent connections, which we have named *GMPLS Daemon.*

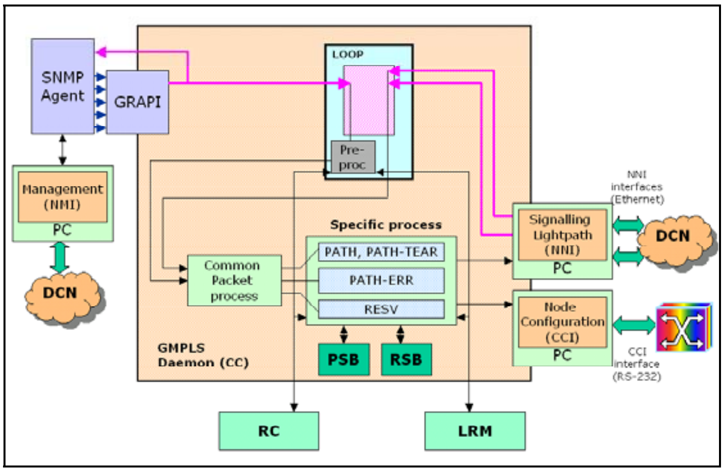


Figure 9. GMPLS Daemon

GMPLS Daemon is based on the RSVP protocol for optical channel connection signalling, and consequently a great part of RSVP operation mechanisms have been adopted. This implementation specifies a subset of RSVP messages and objects that are considered fundamentals for the NNI signalling from the point of view of establishing SPCs, according to GMPLS extensions introduced by [6]. A specific Application Program Interface (API) for GMPLS-RSVP, which we have named *GMPLS RSVP* API (GRAPI), has been implemented allowing local communication with *GMPLS Daemon*. The GRAPI interface is based on a client library, whose functions collect all the GMPLS-RSVP parameters needed for SPC. The GRAPI has been especially designed for being used by the SNMP agent of the optical node (Figure 13), which interprets the queries and commands sent by the NMS for setup and teardown of SPC and calls the corresponding GRAPI functions. SNMP agent's commands allow to set up soft-permanent uni- or bi-directional optical channels, with wavelength-continuity constraint, specifying the following attributes: source *node_id*, *tunnel_id*, *lsp_id*, *extended_tunnel_id* (defining a session for the optical connection), *source_client_id*, *source_port_client_id* (identifying the client attachment point of the source), *destination_node_id*, *destination_port_id* (identifying the destination OCC and port), *lsp_encoding_type*, *switching_type*, *generalized_pid* (identifying the payload) and other traffic parameters such as the peak data rate. The communication between the GRAPI and the *GMPLS Daemon* is performed through a Unix socket that is requested to the *GMPLS Daemon* every time the GRAPI is initialised. This socket is also used by the SNMP agent to listen to events sent by the *GMPLS Daemon*, such as errors or requested connection acknowledgments, which will be used as response messages, notifications or alarms to be sent to the NMS.

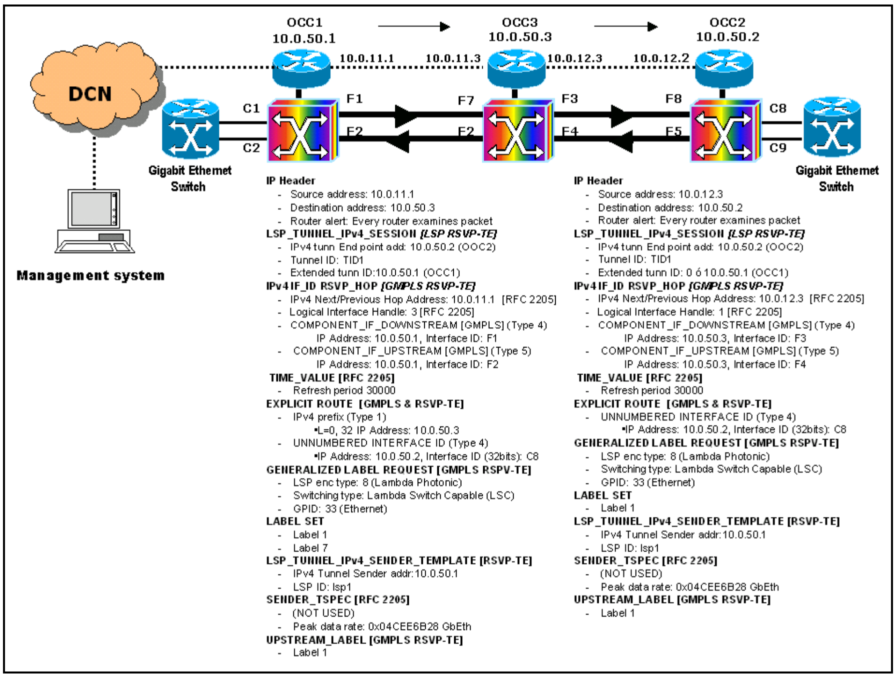
The *GMPLS Daemon* has three main functions: listening, processing and packet delivery (performed by PC). The first function consists of a main loop that listens to all the established sockets until one of them is activated. Apart from the socket established with the GRAPI (SNMP agent), there are other types of sockets, such as those established with the NNI signalling interfaces, allowing the reception of NNI RSVP messages from other OCCs. Note that the socket established with the UNI interface, allowing the reception of UNI control messages from the TES, is not considered in this paper. So, when a socket is activated, the daemon identifies which socket has been activated to call the function that will process the information contained in the socket.

When the SNMP agent requests an optical connection establishment by means of the GRAPI client library, the GRAPI generates a request packet and sends it to the *GMPLS Daemon* through the established socket, so that as soon as the *GMPLS Daemon* advertises the activation of this socket, it can call a pre-processing function that maps the parameters of the received information into Path message parameters, specifying in the *previous_hop* field a value identifying that this packet comes from the GRAPI and not from the network. Moreover, it requests to the RC a strict path between the source node and the destination node, and to the LRM a set of available wavelengths. The strict path is mapped into a Explicit Route object, specifying, for the destination OCC, the *destination_port_id* using a new subject of type 4 that specifies unnumbered links. The rest of OCCs are specified only by their node identifiers using a sub-object of type 1. The set of available wavelengths is mapped into a Label Set object, allowing connections with wavelength continuity, and the client attachment point of source OCC is mapped into an IPv4_IF_ID_RSVP_HOP object.

Finally, the Path message is formed by the following objects: *Session* and *Lsp_Tunnel_IPv4_Sender_Template* to unambiguously identify the G-LSP session, *Generalized_Label_Request* to request a G-LSP based on lambda switching and to indicate the payload type, IPv4_IF_ID_RSVP_HOP to indicate which node and by which fibre (port) is requested the lambda, SENDER_TSPEC to specify the required bandwidth, LABEL_SET to specify a set of labels (lambdas) that constraint label selection downstream, and EXPLICIT_ROUTE, which specifies a strict path and destination port identifier. Another object that can be included in the Path message is UPSTREAM_LABEL to allocate a label on downstream signalling, allowing bi-directional optical connections.

Then, a common processing function is called in order to do the same process to both Path messages coming from the NNI signalling interface and Path messages coming form the GRAPI. Relating to common processing, the first step is to check packet type (Path, Resv, PathTear or PathError. Note that from the perspective of the ASON model ResvErr and ResvTear messages are not used) to perform a specific process. In case of Path message (Figure 10), we check the possibility of an associated session for this message, and, if there is one, the message is a refresh and therefore only its associated *Path State Block* (PSB) must be updated in case of connection parameters' modification. If there is no associated session, the received Path message is requesting an optical connection, and therefore a new PSB must be created, containing all the optical connection attributes. Then, all the objects included in the Path message are analysed in order to check the feasibility of the requested optical connection through the information obtained from the RC and LRM. If the requested connection is not feasible, a PathError message with the *Path_State_Removed* flag set (allowing state to be removed by intermediate nodes) is generated to the upstream node, indicating the error type, whereas if the requested connection is feasible, the LRM sets the wavelength status of all the available wavelengths that match the Label Set to *reserved*. If the requested connection is bi-directional, the

LRM also reserves the received upstream wavelength. After that, the neighbour node is looked up for Path message's next hop in Explicit Route object, and the fibre by which to request the wavelength is looked up in the LRM (mapped to IPv4_IF_ID_RSVP_HOP object). If the requested connection is bi-directional, the LRM reserves an available wavelength that must be the same as the Upstream Label received previously, in order to satisfy the wavelength-continuity constraint. This label will be sent downstream, mapped into the Upstream Label object. Finally, the LRM provides a new Label Set composed by all the wavelengths that previously matched the reserved Label Set, and a new Path Message with all the new objects is sent to the next node. Note that reserved resources and objects sent to the next Path message are stored in the PSB but no physical switching is done at this stage.



IP Header
- Source address: 10.0.11.1
- Destination address: 10.0.50.3
- Router alert: Every router examines packet

LSP_TUNNEL_IPv4_SESSION *[LSP RSVP-TE]*
- IPv4 tunn End point add: 10.0.50.2 (OOC2)
- Tunnel ID: TID1
- Extended tunn ID:10.0.50.1 (OCC1)

IPv4 IF_ID RSVP_HOP *[GMPLS RSVP-TE]*
- IPv4 Next/Previous Hop Address: 10.0.11.1 [RFC 2205]
- Logical Interface Handle: 3 [RFC 2205]
- COMPONENT_IF_DOWNSTREAM [GMPLS] (Type 4)
  IP Address: 10.0.50.1, Interface ID: F1
- COMPONENT_IF_UPSTREAM [GMPLS] (Type 5)
  IP Address: 10.0.50.1, Interface ID: F2

TIME_VALUE [RFC 2205]
- Refresh period 30000

EXPLICIT ROUTE [GMPLS & RSVP-TE]
- IPv4 prefix (Type 1)
  - L=0, 32 IP Address: 10.0.50.3
- UNNUMBERED INTERFACE ID (Type 4)
  - IP Address: 10.0.50.2, Interface ID (32bits): C8

GENERALIZED LABEL REQUEST [GMPLS RSPV-TE]
- LSP enc type: 8 (Lambda Photonic)
- Switching type: Lambda Switch Capable (LSC)
- GPID: 33 (Ethernet)

LABEL SET
- Label 1
- Label 7

LSP_TUNNEL_IPv4_SENDER_TEMPLATE [RSVP-TE]
- IPv4 Tunnel Sender addr:10.0.50.1
- LSP ID: lsp1

SENDER_TSPEC [RFC 2205]
- (NOT USED)
- Peak data rate: 0x04CEE6B28 GbEth

UPSTREAM_LABEL [GMPLS RSVP-TE]
- Label 1

IP Header
- Source address: 10.0.12.3
- Destination address: 10.0.50.2
- Router alert: Every router examines packet

LSP_TUNNEL_IPv4_SESSION *[LSP RSVP-TE]*
- IPv4 tunn End point add: 10.0.50.2 (OOC2)
- Tunnel ID: TID1
- Extended tunn ID: 0 ó 10.0.50.1 (OCC1)

IPv4 IF_ID RSVP_HOP *[GMPLS RSVP-TE]*
- IPv4 Next/Previous Hop Address: 10.0.12.3 [RFC 2205]
- Logical Interface Handle: 1 [RFC 2205]
- COMPONENT_IF_DOWNSTREAM [GMPLS] (Type 4)
  IP Address: 10.0.50.3, Interface ID: F3
- COMPONENT_IF_UPSTREAM [GMPLS] (Type 5)
  IP Address: 10.0.50.3, Interface ID: F4

TIME_VALUE [RFC 2205]
- Refresh period 30000

EXPLICIT ROUTE [GMPLS & RSVP-TE]
- UNNUMBERED INTERFACE ID (Type 4)
  - IP Address: 10.0.50.2, Interface ID (32bits): C8

GENERALIZED LABEL REQUEST [GMPLS RSPV-TE]
- LSP enc type: 8 (Lambda Photonic)
- Switching type: Lambda Switch Capable (LSC)
- GPID: 33 (Ethernet)

LABEL SET
- Label 1

LSP_TUNNEL_IPv4_SENDER_TEMPLATE [RSVP-TE]
- IPv4 Tunnel Sender addr:10.0.50.1
- LSP ID: lsp1

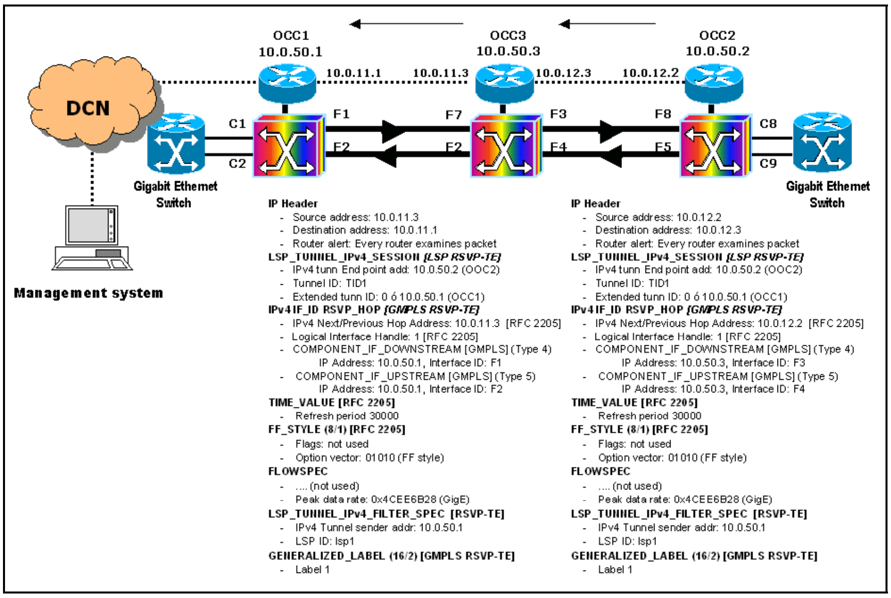
SENDER_TSPEC [RFC 2205]
- (NOT USED)
- Peak data rate: 0x04CEE6B28 GbEth

UPSTREAM_LABEL [GMPLS RSVP-TE]
- Label 1

Figure 10. Path messages

When the Path message reaches its destination, the last OCC responds with a Resv Message (Figure 11) that includes, apart from session identifiers and traffic specifications' objects, the GENERALIZED_LABEL object, which contains a single wavelength for the requested lightpath, chosen by the LRM from previously reserved wavelengths using the First-Fit algorithm. The LRM changes the wavelength status to *in-service* for the selected wavelength, and to *stand-by* for the rejected wavelengths. If the requested connection is bi-directional, the LRM changes as well the status of the upstream wavelength to *in-service*. Moreover, the optical node is configured in order to connect the reserved wavelength with the destination port identifier define previously by the Explicit Route object. Analogously to the Path messages, a Resv State Block (RSB) associated to its respective PSB is created to collect all the information about reserved resources.

This Resv message is sent to the source node following the same path of the associated Path messages. The common processing for Resv messages in intermediate nodes is the same as described above, in which the received Generalized Label object indicates the wavelength that must be selected, and consequently the rest of reserved wavelengths must be freed. A new Resv message, with exactly the same wavelength, must be sent to the previous hop. Obviously, the optical hardware needs to be configured in order to provide the optical channel. Note that the Generalized Label and port identifiers in IPv4_IF_ID_RSVP_HOP are based on local identifiers, so it is necessary that each node that receives a message with these objects map these identifiers to values locally known. If any intermediate node has any problem and cannot continue, it must send a PathTear message downstream and a PathErr (with Path_State_Removed flag set) message upstream. When a Resv message reaches the source node, the SNMP agent informs the NMS about the success of its connection request.



IP Header
- Source address: 10.0.11.3
- Destination address: 10.0.11.1
- Router alert: Every router examines packet

LSP_TUNNEL_IPv4_SESSION *[LSP RSVP-TE]*
- IPv4 tunn End point add: 10.0.50.2 (OOC2)
- Tunnel ID: TID1
- Extended tunn ID: 0 ó 10.0.50.1 (OCC1)

IPv4 IF_ID RSVP_HOP *[GMPLS RSVP-TE]*
- IPv4 Next/Previous Hop Address: 10.0.11.3 [RFC 2205]
- Logical Interface Handle: 1 [RFC 2205]
- COMPONENT_IF_DOWNSTREAM [GMPLS] (Type 4)
  IP Address: 10.0.50.1, Interface ID: F1
- COMPONENT_IF_UPSTREAM [GMPLS] (Type 5)
  IP Address: 10.0.50.1, Interface ID: F2

TIME_VALUE [RFC 2205]
- Refresh period 30000

FF_STYLE (8/1) [RFC 2205]
- Flags: not used
- Option vector: 01010 (FF style)

FLOWSPEC
- .... (not used)
- Peak data rate: 0x4CEE6B28 (GigE)

LSP_TUNNEL_IPv4_FILTER_SPEC [RSVP-TE]
- IPv4 Tunnel sender addr: 10.0.50.1
- LSP ID: lsp1

GENERALIZED_LABEL (16/2) [GMPLS RSVP-TE]
- Label 1

IP Header
- Source address: 10.0.12.2
- Destination address: 10.0.12.3
- Router alert: Every router examines packet

LSP_TUNNEL_IPv4_SESSION *[LSP RSVP-TE]*
- IPv4 tunn End point add: 10.0.50.2 (OOC2)
- Tunnel ID: TID1
- Extended tunn ID: 0 ó 10.0.50.1 (OCC1)

IPv4 IF_ID RSVP_HOP *[GMPLS RSVP-TE]*
- IPv4 Next/Previous Hop Address: 10.0.12.2 [RFC 2205]
- Logical Interface Handle: 1 [RFC 2205]
- COMPONENT_IF_DOWNSTREAM [GMPLS] (Type 4)
  IP Address: 10.0.50.3, Interface ID: F3
- COMPONENT_IF_UPSTREAM [GMPLS] (Type 5)
  IP Address: 10.0.50.3, Interface ID: F4

TIME_VALUE [RFC 2205]
- Refresh period 30000

FF_STYLE (8/1) [RFC 2205]
- Flags: not used
- Option vector: 01010 (FF style)

FLOWSPEC
- .... (not used)
- Peak data rate: 0x4CEE6B28 (GigE)

LSP_TUNNEL_IPv4_FILTER_SPEC [RSVP-TE]
- IPv4 Tunnel sender addr: 10.0.50.1
- LSP ID: lsp1

GENERALIZED_LABEL (16/2) [GMPLS RSVP-TE]
- Label 1

Figure 11. Resv messages

When the SNMP agent requests an optical connection deletion by means of the GRAPI client library, the same mechanism described above for the Path message is implemented. The PathTear is sent from the source node to the destination node and deletes both Path and Resv states for the associated connection. Note that this RSVP NNI signalling implementation does not use a reliable delivery mechanism for RSVP messages such as [7], so if a RSVP packet requesting or deleting a connection is lost, the only available protection mechanism is refresh (in 30 seconds the packet will be re-sent).

3.5 SNMP AGENT

In this paper we consider a generic management model, compliant with the ITU-T Telecommunication Management Network (TNM) approach, for setup and teardown of soft-permanent optical connections. This model is based upon the manager-agent paradigm, being its key elements in an ASON framework [1] (i) the optical Network Management System (NMS), located in the ASON management plane and acting as an information system manager for network control and monitoring, (ii) the agents, located in the optical nodes of the ASON control plane (OCCs), (iii) the management protocol, upon which is based the Network Management Interface (NMI, Figure 4) and (iv) Management Information Bases (MIB), which are abstractions of the optical network's management information consisting of network resources admitting management (managed objects). Due to the requirements of an ASON, in this paper we consider MIB standards for TCP/IP-based networks [8] and an NMI based upon SNMP (Figure 12b) in line with IETF SNMP Management Framework.

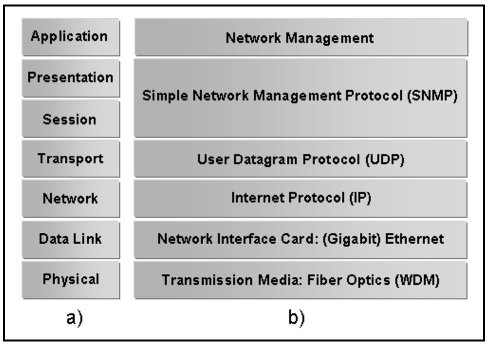


Figure 12. a) OSI layers and b) protocol implementation for ASON

For setup and teardown of SPC, the above-mentioned key elements interact with each other as follows. The manager (NMS) has a mapping of the MIBs allocated in the agents (OCCs), and sends management queries and commands to the agents through the NMI via the DCN, whereas the SNMP agents store and update their MIBs, answer management queries and send notifications and alarm messages to the NMS. For this purpose, each agent located in an OCC is an SNMP agent with the functional architecture depicted in Figure 13, which consists of (i) management applications, (ii) data transport services and (iii) management support functions.

As for management applications, from the generic management functional areas (FCAPS) [9] we consider configuration management of resources related to soft-permanent optical connections, that is, GMPLS labels, as described in Section 3.4. Thus, management applications are *Notification Originator* and *Lightpath Setup/Teardown Responders* (Figure 13), being each SNMP agent responsible for replying the soft-permanent optical connection setup and teardown requests triggered by the NMS through access and update of GMPLS labels management information stored in its MIB. Note that in the considered framework the objective of configuration management is the provision of end-to-end soft-permanent optical connectivity, whose setup and teardown is triggered by means of the NMS and is provided using the control plane mechanisms described in Section 3.4.

Data transport services are located in the application layer (Figure 12a) and are intended to pack and unpack protocol data units used in SNMP primitives such as Get, Set or Trap. Last but not least, management support functions are MIB access and update modules, as well as SNMP communications protocol stack (UDP/IP layers, depicted in Figure 12b). For both levels, in this paper we consider an SNMP Linux API (Figure 13). An additional data transport service is GRAPI library, devoted to communication between the SNMP agent and the *GMPLS Daemon*, as described in Section 3.4.

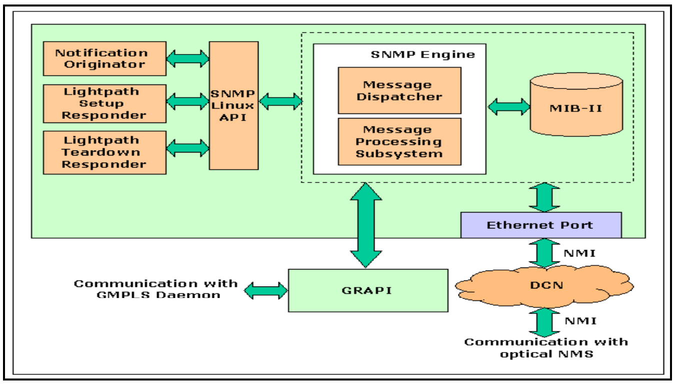


Figure 13. SNMP agent functional architecture

Managed objects are labelled in the Internet registration hierarchy tree (ITU-T X.660) under the level with object identifier .1.3.6.1.4.1.13909. SNMP agent's MIB philosophy is in line with the requirements of GMPLS management and the strategies for modelling and managing labels within GMPLS systems [10]. Note that, due to *GMPLS Daemon* architecture, there is no need for a TE MIB in the optical nodes, hence managed objects considered for setup and teardown of optical connections are the GMPLS labels characterizing a soft-permanent connection, that is, destination OCC, traffic type, payload, bi-directionality and continuity options, explicit route and switching type, as well as the lightpath identifiers TUNNEL_ID and EXTENDED_TUNNEL_ID defined by the optical NMS.

For each setup request triggered by the optical NMS, the SNMP agent of the source OCC performs a connection setup request to its *GMPLS Daemon* through GRAPI, as described previously. In case of successful establishment of the optical connection, according to the mechanisms described in Section 3.4, the *Ligthpath Setup Responder* of the SNMP agent updates the managed objects characterizing the established connection and sends a successful response message to the optical NMS. For each teardown request triggered by the optical NMS, the SNMP agent of the source OCC performs a connection teardown request to its *GMPLS Daemon* trough GRAPI. In case of successful teardown of the optical connection, the *Ligthpath Teardown Responder* of the SNMP agent erases the managed objects characterizing the deleted connection and sends a successful response message to the optical NMS.

In case of impossibility to perform a soft-permanent request, the SNMP agent's *Lightpath Setup*/*Teardown Responder* sends an error message to the optical NMS containing the cause of the error and, in case of a failure or an event not triggered by the optical NMS, the SNMP agent sends an alarm or a notification message to the NMS via its *Notification Originator*.

4. CONCLUSIONS

Summarizing, this paper presents the optical control plane architecture implemented on a real ASON/GMPLS testbed for setup and teardown of soft-permanent connections, focusing on design and implementation issues of the Data Communication Network and the Optical Connection Controllers, specifying the modules, protocols and mechanisms such as network topology discovery (we consider distributed adaptive routing based on global information and implemented on OSPF link-state routing protocol), available optical resources (we have defined a Link Resource

Table that contains all the information about fibres and wavelengths locally available, using an addressing scheme based on Unnumbered Links and Bundled Links), route computation and wavelength allocation (using Dijkstra's algorithm to choose the shortest cost path and First-Fit algorithm to assign the wavelength) and ligthpath setup (using RSVP-TE as NNI signaling protocol with extensions for GMPLS).

In the future, we will implement more effective weight functions and wavelength allocation algorithms in order to compare them and to evaluate their performance in terms of blocking probability, link utilization or average delay. Moreover, we will perform the dissemination of the network's local optical resources in order guarantee the wavelength-continuity constraint.

ACKNOWLEDGEMENT

This work is part of the project EMPIRICO, partially supported by the Spanish Government under contract PU-2002-56.

REFERENCES

[1] ITU-T Recommendation G.8080, *Architecture for the Automatically Switched Optical Network (ASON)*, November 2001.

[2] E. Mannie et al, *Generalized Multi-Protocol Label Switching architecture*, IETF draft-ietf-ccamp-gmpls-architecture-* (work in progress).

[3] O. Abul-Magd et al., *Interim User Network Interface (UNI) signaling specification*, OIF2000.125.7, April 2001.

[4] H. Zang, P. Jue, B. Mukherjee, *A review of Routing and Wavelength Assignment approaches for wavelength-routed optical WDM networks*, Optical Networks Magazine, Vol. 1, Num. 1, pp. 47-60, January 2000.

[5] T. Fabry-Asztalos, N. Bhide, K.M. Sivalingam, *Adaptive weight functions for Shortest Path routing algorithms for multi-wavelength optical WDM networks*, IEEE ICC 2000, New Orleans (USA), 18-22 June 2000.

[6] P. Ashwood-Smith et al., *Generalized MPLS signaling - RSVP-TE extensions, IETF* draft-ietf-mpls-generalized-rsvp-te-* (work in progress).

[7] L. Berger et al, *RSVP refresh overhead reduction extensions*, IETF RFC 2961, April 2001.

[8] K. McCloghrie, M. Rose, *Management Information Base for network management of TCP/IP-based internets: MIB-II*, IETF RFC 1213, 1991.

[9] O. Aboul-Magd et al., *Automatic Switched Optical Network architecture and its related protocols*, IETF draft-ietf-ipo-ason-* (work in progress).

[10] T. Nadeau et al., *Generalized Multiprotocol Label Switching label Management Information Base*, IETF draft-nadeau-ccamp-gmpls-label-mib-* (work in progress).