An Optical UNI Architecture for the GIGA Project Testbed Network

Rafael Pasquini, *Student, IEEE*, Fábio L. Verdi, *Student, IEEE*, Luiz Gustavo Zuliani, *Student, IEEE*, Maurício Magalhães, *Member, IEEE* and Sandro Rossi, *Member, IEEE*

***Abstract*— The main interests of the GIGA Project are the deployment of Optical Network Technology, Network Services and Applications, Experimental Telecommunication Services and Scientific Applications. In the context of this Project, our research group has been working in the design of a IP/WDM Control Plane to integrate the IP/MPLS Client Network with the Optical Transport Network (WDM), according to the GMPLS and ASON specifications. We propose a UNI (User Network Interface) architecture providing an independent signaling protocol to integrate the Client and the Optical Transport Networks. The proposed UNI architecture maintains unchanged the client and transport networks signaling semantics for the GIGA testbed network. The paper describes how the designed architecture supports the independence of the signaling protocol and presents the prototype implemented to validate the architecture.**

***Index Terms*— GMPLS, UNI, Overlay model, signaling protocols, optical networks.**

I. INTRODUCTION

The strong growth in the Internet traffic and services and the new optical transmission and switching technologies were the motivations that converged to the implementation, in May 2004, of the GIGA Project testbed network. The network has 735km and interconnects 17 universities and research centers in the Rio-São Paulo axis. It is an experimental network aiming at deploying of optical network technologies, applications and services related to IP and broadband networks. The network has links with 2,5 Gbps that in the future would reach 10 Gbps in each wavelength. The technology used is WDM (Wavelength Division Multiplexing) [1]. This technology associates optical signals to different wavelengths (lambdas), allowing the separation of data channels in the same fiber. The project was divided in to several research groups. Our group have been working in collaboration with other institutions [1] on the deployment of the experimental network IP control plane. This control plane must provide actions automatically such as the establishment and teardown of lightpaths, and fault recovery.

The most promising technology able to meet these requirements is the GMPLS protocol suite defined by IETF [2]. GMPLS extends the label-switching paradigm introduced by MPLS to network elements that have non-packet-based forwarding engines. The GMPLS protocol suite specifies signaling [3], routing [4] and link management [5] procedures.

Besides the protocols constituting the control plane, a functional architecture is also required. In order that the ITU-T deployed a functional architecture named ASON (Automatically Switched Optical Network) [6] which specifies the components that must be present in the control plane and the functional requirements that an optical transport network should have. This architecture uses the Overlay model to interconnect the client and transport networks. Under this model, the client network has no visibility of the transport network core, i.e., the transport network is seen as a "black box" from the client perspective. The Overlay model needs a UNI interface to interconnect the client and transport networks. Currently, there are two UNI specifications, one proposed by OIF [7] and another one by IETF [8].

An important UNI characteristic is its independence of the signaling protocol since the UNI specification defines only abstract messages for the interaction between the client and transport networks. This characteristic allows the existence of different signaling protocols in the client and in the transport networks. The signaling protocol specified for the UNI implementation in the GIGA Project was based on RSVP. The RSVP [9] was originally deployed to reserve resources on routers belonging to Integrated Services capable networks. This protocol works with messages composed by variable-sized objects. As new objects can be included in the RSVP messages, it can be easily upgraded. Considering this facility, it was extended to work within the MPLS context. Basically, it received some new objects to provide Traffic Engineering mechanisms, e.g., an object to specify alternative routes (*ERO*) and objects to reserve MPLS labels (*LABEL and LABEL_REQUEST*). As a consequence, it is now called RSVP-TE [10]. However, the RSVP-TE does not operate with technologies that deal with wavelength, fiber and time switching. To address this issue, it was updated again to be compatible with GMPLS and is named GMPLS RSVP-TE [3], [11]. Under the GIGA Project the client network uses RSVP-TE (MPLS) and the optical transport network will support GMPLS RSVP-TE.

This work proposes a UNI architecture that maintains unchanged the client and transport network signaling. The architecture follows the specifications recommended by the entities mentioned before and proposes an independent sig-

---

Manuscript created March 30, 2006. This work was supported by GIGA Project and its partners, Brazil.

R. Pasquini (e-mail: pasquini@dca.fee.unicamp.br), F. Verdi (email: verdi@dca.fee.unicamp.br), L. G. Zuliani (e-mail: zuliani@dca.fee.unicamp.br) and M. F. Magalhães (e-mail: mauricio@dca.fee.unicamp.br) are with the Department of Computer Engineering and Industrial Automation (DCA), School of Electrical and Computer Engineering (FEEC), State University of Campinas (UNICAMP), Campinas-SP, Brazil (DCA-FEEC-UNICAMP. C. P. 6101. Campinas-SP. Brazil. CEP 13081-970). S. Rossi (e-mail: sandro@cpqd.com.br) is with CPqD Telecom & IT Solutions, Campinas-SP, Brazil (Campinas-Mogi Mirim road - km 118,5. Campinas-SP. Brazil. CEP 13086-902)

[1] Atlantic Institute and CIN/UFPE.

naling protocol UNI to attend the characteristics of the GIGA Project as well as other networks implementing the Overlay or Augmented models. The architecture allows the mapping from any signaling protocol existent in the client network to UNI API invocations. So, the architecture has one level of abstraction allowing the UNI to work in environments with different signaling protocols.

The next sections of this paper are organized as follows: Section II presents the network interconnection models available (Overlay, Peer and Augmented); Section III introduces a comparison between the UNI specifications done by OIF and IETF; the proposed UNI Architecture is presented in Section IV; in Section V is possible to see how the proposed Architecture is implemented in the GIGA Project; the deployed prototype and some results obtained from simulations are shown in Section VI and, closing the paper, the Section VII brings the conclusions and some possible future works.

II. CONTROL PLANE INTERCONNECTION MODELS

An efficient interaction between the client network and the transport network is indispensable. Currently, there are different specifications with different cooperation levels to address this interaction. The Overlay and Peer are opposite models whereas the Augmented Model can be seen as an intermediary one. This section presents these control plane interconnection models and their relationships with the UNI.

A. Overlay Model

The Overlay Model assumes a total separation between the client and transport network control planes. The model limits the signaling information exchanging, i.e., the signaling and routing protocols inside the client network are totally independent of the protocols inside the transport network. The signaling interaction between both control planes occurs through the UNI in a client-server relationship. As a consequence, this model is the most opaque and has less flexibility than the Peer and Augmented models.

Despite the fact that this model has some interaction limitations, it is very common among the carriers, mainly because of the offered opacity allowing the carriers to “hide” the backbone topology from their clients.

B. Peer Model

The Peer Model assumes, in opposition to the Overlay Model, a trust-full relationship between the domains. As an example, we can mention a carrier that has both the transport and access services and wants to optimize its topological alignments. The client network control plane and the transport network control plane are peers. So, just one control plane instantiation runs over both networks, i.e., this model does not need to use a UNI.

Carriers restriction to this model is due to the diffusion of their internal topology to other domains since most of them do not admit this possibility.

C. Augmented Model

In terms of trustability, the Augmented model can be considered as an intermediary model compared to the Overlay and Peer models. The UNI is necessary in the Augmented Model to allow the exchange of limited quantity of routing information between the client and transport networks. This is the characteristic which makes this model different from the Peer model in which complete routing information is exchanged between the networks. The Augmented model allows basically the exchange of reachability information between the edge elements in the client and transport networks.

Another difference between the models is relative to the address space used. While the Peer model shares the same address space, the Overlay model uses disjoint address spaces between the client and transport networks. Under the Augmented Model, the same address space utilization is allowed, but not mandatory. Such mechanism controls the transport network opacity level.

III. OIF UNI X IETF UNI

Currently, there are two UNI specifications, one made by OIF (also called Public UNI) [7] and another one made by IETF (also called Private UNI) [8], both are GMPLS-compliant. The OIF UNI was deployed to work with the Overlay model. It is ideal for networks composed of different administrative domains. The IETF UNI is ideal for environments with the same administrative domain and offers more possibilities and functionalities than the OIF UNI. It works with the Augmented model and it is also possible to support the Overlay model such as the Public UNI does.

The Public UNI assumes a strict separation of the client network control plane and the transport network control plane. Because of the unreliable characteristic between the administrative domains inherent to the Overlay model, the Public UNI does not exchange reachability and topological informations. It has only messages to establish, tear down and query the status of the established paths in the transport network. The requests done to the transport network follows a pre-established service level agreement (SLA). The control planes separation is, at the same time, its better and worse characteristic. On one side, it limits the quantity of control information that must be exchanged. On the other side, as the transport network is seen as a “black box” to the client, few coordination is allowed between the involved networks. As a consequence, there is an overlapping of signaling, routing and traffic engineering functions. In other words, it means that great effort is necessary to optimize the offered transport network resources, especially when there is an intense path reconfiguration due to traffic dynamics and network faults.

Although the Public UNI is GMPLS-based, it defines some OIF-specific extensions that are not GMPLS-compliant. For example, the *Generalized UNI Object*. To maintain the client and transport networks independence and the signaling protocol transparency, the Public UNI establishes three different sessions in the transport network. The first one is established between the ingress edge client node and the ingress edge core node. The second session is established between the core transport network elements. The third session is established between the egress edge core node and the egress edge client node. What define a session are the destination IP address and the path ID of the requested path. Under Public UNI, all the three session addresses are different from the end-to-end client session. In this case, it is impossible to determine the egress transport network node address where the path will finish. The *Generalized UNI Object* solves this problem transporting the client session address during the path establishment phases.

Other extensions are needed to the Public UNI works properly on the actual scenarios. With this in mind, the IETF started to define its own full GMPLS-compliant UNI (Private UNI). Because of the fact that this new interface is based on GMPLS protocol suite and can work with the Augmented model or with the Overlay model, it is possible to have a bigger and cleaner interaction between the involved network control planes. Although it was specially defined to work on environments with the same administrative domain, it is possible for it to work on different administrative domains as done by Public UNI. This IETF UNI is a RSVP-based interface. If the transport network implements the same protocol that is implemented by UNI (RSVP), just one session is needed. As a consequence, it simplifies the signaling process inside the transport network. On the other hand, if there are different signaling protocols between the transport network and the UNI, three sessions are needed as in the OIF UNI.

The Private UNI has some new features such as the possibility to change parameters on established paths and multilayer fault recovery. This last new feature, basically offers mechanisms for fault notification and to establish alternative paths in response to network failures.

IV. UNI ARCHITECTURE PROPOSAL

The UNI function is to provide the interaction between the client network and the transport network. However, it is possible to suppose that these networks have different signaling protocols. As a consequence, the UNI implementation requires a great effort. With this in mind, a signaling protocol independent UNI architecture is proposed in this paper. At the same time, this architecture is totally transparent to the client network since the client does not need to know about the UNI existence.

It is important to mention that there are some UNI related works as [12] and [13] that discuss the UNI implementation and fault recovery in scenarios with the same administrative domain in which the client network must knows about the UNI existence.

Currently, there are many signaling protocols as CR-LDP, RSVP-TE and GMPLS RSVP-TE. The UNI specification defines some abstract messages. These messages are called abstract because their implementations depends of the chosen signaling protocol. For the proposed architecture, we defined a signaling protocol based on the GMPLS RSVP-TE. The protocol state machine is similar to that one representing GMPLS RSVP-TE, however, the protocol data units are codified as XML messages. This option is to facilitate the extensions of the signaling protocol to include new functionalities to the UNI. The price paid by this option is the large generated messages due to the fact that the XML codification is based on ASCI II characters.

The proposed architecture is presented in Figure 1.

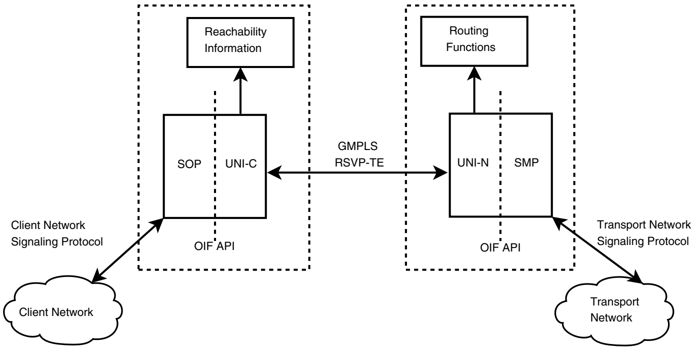

Fig. 1. Proposed UNI Architecture.

The UNI has two modules, UNI-C (Client) and UNI-N (Network), which under the proposed architecture communicate using the GMPLS RSVP-TE-like signaling protocol. The UNI-C, as specified by OIF, has a standard interface that must be invoked by the client network when it wants to establish a new lightpath (LSP) through the transport network, i.e., in a normal UNI implementation, the client must knows that the UNI exists and more, knows the UNI-C API to be able of requesting connections through the transport network. To solve this problem the proposed architecture uses a new module called SOP (Signaling Overlaying Point).

The SOP module acts as a client signaling adapter. It receives all the client messages that reach the transport network border and maps these messages to UNI-C API invocations. As the client networks can use different signaling protocols, it is possible to have more than one SOP attached to one UNI-C mapping each signaling protocol. The SOP retains the client message on its interior until the LSP through the transport network is being established. Once the LSP is established, the SOP tunnels the client message in the recently established LSP. As a result, the SOP maintains the client signaling semantic totally preserved and the client does not need to know about the UNI.

The UNI-C performs the admission control procedures such as: verify if the client is allowed to request resources from the transport network (based on policies or SLAs); check the possibility of offering the requested resources to the client and verify if the transport network reaches the destination required by the client (Reachability Information module).

The UNI-N inside the transport network receives the UNI-C requests and asks to the routing (Routing Functions module) a path through the transport network that reaches the destination node. When the UNI-N receives the route, it triggers the signaling inside the transport network through the SMP (Signaling Mapping Point) module.

The SMP module is used to adapt the signaling between the UNI and the transport network. Using the SMP module, it is possible to use this UNI architecture in scenarios with different signaling protocols and maintain not only the UNI but also the transport network signaling protocol semantics unchanged.

For the GIGA Project it is assumed the use of GMPLS RSVP-TE in the transport network control plane. The communication between UNI-N and SMP occurs through the OIF UNI API specification.

V. USING THE PROPOSED ARCHITECTURE IN THE GIGA PROJECT TESTBED NETWORK

The GIGA Project will have high performance PCs in a one-to-one relation with each OxC inside the transport network. The client network will employ commercial MPLS routers which do not admit any kind of modifications as, for example, updating of the MPLS embedded software or even the inclusion of new functionalities.

The interconnection model adopted by the Project was the Overlay Model which demands, therefore, a UNI to support the interaction between the client and transport networks. As discussed earlier, the UNI-C is implemented in the client network and the UNI-N in the transport network. Despite of the fact that it is impossible to make changes in the client network equipments, the UNI-C will be implemented in a high performance PC situated at the external transport network border. This high performance PC will have optical interfaces that will be linked to the transport network OxCs interfaces. The forwarding plane has many optical fiber pairs in which some different wavelengths are multiplexed. One of these wavelengths will be used to provide the control plane connectivity (*Out-of-band In-fiber*).

An important point in the GIGA Project is the deployment of platform and manufacturer independent optical technologies. Taking this in to consideration, the proposed architecture uses only protocols that are specified by IETF and OIF. Figure 2 shows how the proposed UNI architecture is implemented in the Project.

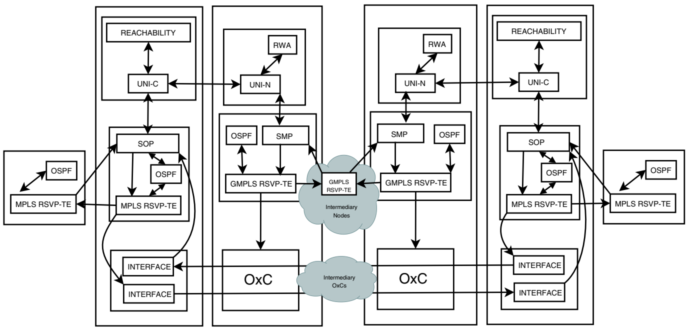

Fig. 2. GIGA Project Implementation of the Proposed Architecture.

The client network signaling protocol is the RSVP-TE for MPLS. The transport network must be totally transparent to the client network. To make this possible, the SOP module is used. When the SOP module receives the client signaling message, it makes a routing table query and based on this query it can: 1) maps the client signaling message in a UNI API invocation establishing a new LSP through the transport network or 2) tunnels this signaling message in one existent LSP. In the first case, when the SOP receives a client message, it stores and maps this message to an UNI API invocation to establish a new LSP. Once the new LSP is established, the two client edge nodes become adjacent since this new LSP is seen as a simple TE link between them, called Forwarding Adjacency. Then, the SOP receives the confirmation message from UNI-C, retrieves the stored client message and tunnels it in the new established LSP maintaining the client totally unaware about the whole process and preserving the client signaling protocol semantic. The second case occurs because of the *softstate* implemented by RSVP-TE protocol. It means that in some defined time intervals, the signaling states created in each router belonging to the client network path, need to be refreshed. This refreshing is done exchanging RSVP *PATH* and *RESV* messages. In this case, the LSP (TE link) is already established and the SOP only needs to tunnel these messages to preserve the client semantic.

When the source node wants to remove one of its connections, it sends a *PATHTEAR* message. This message reaches the SOP that: 1) maps this message in a UNI API invocation to teardown the related LSP and 2) at the same time, it tunnels this message within the LSP to effectively teardown the end-to-end client network connection. This is necessary to keep the client network RSVP session semantic preserved. Note that if other signaling protocol was used in the client network, another behavior would exist. To address this issue, it is only necessary to change the SOP.

The UNI-C is responsible for admission control procedures. Among these procedures, the UNI-C verifies if the requested destination is reachable by the transport network. In the GIGA Project, this information is obtained using a manually filled up reachability table (Reachability Information module). If any of these procedures finishes on error, the UNI-C informs what happened to the SOP which translates this message to the client signaling protocol (a *PATHERROR* message in the GIGA Project case).

Once the UNI-C finishes its tasks, it forwards the request to the UNI-N. The last one is responsible for checking the transport network resources availability. The OxCs used in the GIGA Project do not perform wavelength conversion, i.e., the same wavelength must be used on the whole LSP. A RWA (Route and Wavelength Assignment) algorithm is used to find a route and a wavelength inside the transport network. The UNI-N asks the RWA a route and sends this one to the SMP module that triggers the signaling in the transport network control plane.

The proposed architecture uses two sessions inside the transport network. Figure 3 shows these sessions and the client end-to-end session. Session 1 is identified by TE.Y node address and it corresponds to a transport network control plane (GMPLS RSVP-TE) session. Session 2 is identified by CE.Y and corresponds to a session between the involved UNI-Cs. This second session is tunneled in the transport network control plane using the *Generalized UNI Object*. Session 3 is the end-to-end client (C.Y) session that is tunneled within the LSP that is seen as a TE link from the client network perspective.

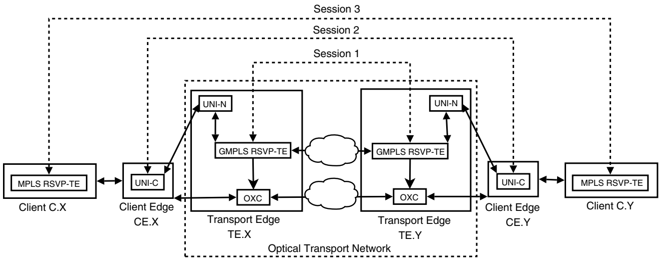

Fig. 3. End-to-end Needed Sessions.

VI. IMPLEMENTATION

After the architecture specification phase, a prototype was deployed to evaluate the behavior and performance of the architecture.

The proposed architecture modules (SOP, UNI-C, UNI-N and SMP) were implemented using JAVA. The used signaling protocol inside the UNI was based on the GMPLS RSVP-TE. This pseudo GMPLS RSVP-TE was implemented also using JAVA. This module uses XML to implement the RSVP equivalent messages and objects. It adopts the same state machine defined to the GMPLS RSVP-TE. The prototype modules are Multithread Socket servers. This characteristic allows the use of different socket ports to simulate big and different topologies even with a small number of computers. Since the modules are Multithread, it is possible to receive several requisitions simultaneously.

After the implementation of the modules, a testbed scenario was created to validate the architecture. To construct the testbed, the GMPLS RSVP-TE-like implemented protocol was used to simulate the transport network control plane signaling functions. The proposed architecture specifies the use of one routing protocol to define a route in the transport network. The prototype includes a simple Routing Server with some static routes that is invoked by the UNI-N module.

Currently, it is possible to manually trigger the establishment and tear down of LSPs through the emulated optical transport network. To better test the SOP module, the GMPLS RSVP-TE-like signaling protocol was modified to be MPLS-compliant and was used to simulate the client network signaling. At the same time, another module (*INTERFACE*) was implemented to simulate the lightpaths (client TE links) where the client signaling messages should be tunneled by the SOP module. So, the SOP module functions (e.g., mapping the client signaling and maintaining the client signaling semantic) could be properly tested. Figure 4 shows the used simulation topology. To deploy this topology, we used 4 Intel®Pentium IV®HT 3.0 GHz with 1 GB of Ram equipped with 4 Fast Ethernet network cards.

The simulation process involves the parsing of XML messages. The XML messages used in the prototype have a considerable size (Table I). This is the price paid for the flexibility introduced in the GMPLS RSVP-TE-like signaling protocol implemented between the UNI-C and UNI-N. Some performance tests were done to verify how these parsing procedures would interfere on the prototype processes. The time performance is shown in two different tables, one with the average times obtained to establish connections (Table II) and another table with the average times to remove connections (Table III).

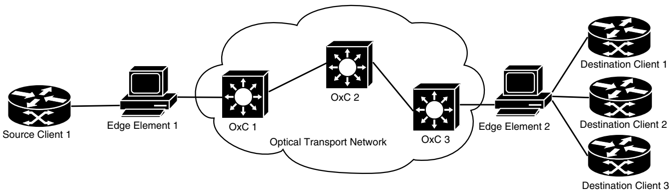

Fig. 4. Simulation Topology.

TABLE I

AVERAGE XML MESSAGES SIZES.

Message	Size (in bytes)
MPLS PATH message	315
GMPLS PATH message	750
MPLS RESV message	300
GMPLS RESV message	430
MPLS PATHTEAR message	280
GMPLS PATHTEAR message	285

TABLE II

AVERAGE TIMES OBTAINED TO ESTABLISH LSPS.

Action	Time (in ms)
End-to-end process	550
SOP - SMP (PATH)	60
SMP - UNI-C (PATH)	30
UNI-C - SMP (RESV)	35
SMP - SOP (RESV)	50

TABLE III

AVERAGE TIMES OBTAINED TO REMOVE LSPS.

Action	Time (in ms)
End-to-end process	355
SOP - SMP (PATH)	60
SMP - UNI-C (PATH)	15
UNI-C - SMP (RESV)	20
SMP - UNI-C (RESV)	15
UNI-C - SMP (PATHTEAR)	60
SMP - UNI-C (PATHTEAR)	35

From these two average time tables it is possible to extract the implemented UNI times, that are 175 ms to establish a LSP and 205 ms to remove a LSP. These are the times that must be considered as results of the proposed architecture evaluation. However, to give a better example of how long it would be necessary to establish a LSP for a real optical network using the proposed UNI Architecture, the reference [14] brings some performance evaluation times for a GMPLS optical network testbed. The related time is the time obtained to establish a LSP using three hops such as in our simulations, that was 561 ms. As a conclusion, using the implemented UNI to establish a connection through the presented optical transport network in [14] would be necessary 736 ms.

The deployed prototype has a GUI (Graphic User Interface) which offers a centralized way to trigger events in the prototype. For example, from the GUI is possible to start up and shut down all the distributed modules used in the simulations. It is also possible to establish and remove LSPs from this GUI. When a LSP is defined and dispatched to be installed, the central GUI generates a MPLS *PATH* message and sends this message to the ingress LSP node. Similarly to the establishment of new LSPs, it is possible to remove LSPs from this GUI. In both cases, all the involved phases in the simulation are shown in a centralized log console that depicts all events that happened in the prototype.

An intelligent functionality was introduced in the prototype to permit more complex and dynamic scenarios. For example, during the establishment of a new LSP it is possible that some failures occur, such as UNI-C admission control failures caused by lack of resources in the transport network or not authorized client requesting for resources. In these cases, *PATHERROR* or *RESVERROR* messages will be generated depending on the simulation phase. For example, in the case of lack of resources, the end-point nodes are allowed to change the quantity of resources requested and retransmit the *PATH* or *RESV* messages with the new values or, in the case where none counter-offer could be done, it generates *PATHTEAR* or *RESVTEAR* messages to remove the established states.

VII. Conclusions and Future Works

We presented a UNI implementation architecture that allows the use of the transport network by the client network equipments that do not need to know about the UNI existence. We look at the proposed architecture from the theoretical and the experimental points of view. The theoretical and experimental results were always consistent.

From the theoretical point of view, the proposed UNI architecture works in the GIGA Project testbed network such as in all other scenarios in which a UNI is needed to perform the interaction between the client network and the transport network. From the experimental point of view, the prototype is an important tool between the specification and the effectively implementation phases. The simulation results show the properly work of the proposed architecture modules.

The next GIGA Project phase is to implement the testbed network control plane. Again, it is needed to stand out the SOP and SMP modules. In case of not having a real UNI implementation, this prototype could be used only by specializing these two modules to work with the signaling protocol existent in the client network and the transport network. It is known that some other changes in the prototype would be necessary to attend some requirements such as the interaction with the RWA algorithm and the use of GIGA specific traffic parameters.

As future works, other scenarios would be investigated using the proposed architecture and the prototype. For example, it is possible to adapt the prototype to work in accordance with the IETF UNI (Private UNI), i.e., using the Augmented Model and establishing just one signaling session inside the transport network. Other desirable functions in the architecture are: 1) the capability to perform multi-layer fault recovery and 2) grooming [15] in the transport network borders to better use the available resources.

Acknowledgments

We would like to thank FAPESP and FINEP/Funttel through GIGA Project for their financial support.

References

[1] G. P. Agrawal, *Fiber-Optic Communication Systems, 3rd Edition*. John Wiley, June 2002.

[2] E. Mannie, “Generalized Multi-Protocol Label Switching (GMPLS) Architecture,” *RFC 3945*, October 2004.

[3] L. Berger, “Generalized Multi-Protocol Label Switching (GMPLS) Signaling Functional Description,” *RFC 3471*, January 2003.

[4] D. Katz, K. Kompella, and D. Yeung, “Traffic Engineering (TE) Extensions to OSPF Version 2,” *RFC 3630*, September 2003.

[5] J. Lang, “Link Management Protocol (LMP),” *RFC 4204*, October 2005.

[6] ITU-T: Architecture for the Automatically Switched Optical Network (ASON), G.8080/Y.1304, 2001.

[7] OIF User Network Interface (UNI) 1.0 Signaling Specification, Release 2: Common Part, February 2004.

[8] G. Swallow, J. Drake, H. Ishimatsu, and Y. Rekhter, “Generalized Multi-Protocol Label Switching (GMPLS) User-Network Interface (UNI): Resource ReSerVation Protocol-Traffic Engineering (RSPV-TE) Support for the Overlay Model,” *RFC 4208*, October 2005.

[9] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, “Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification,” *RFC 2205*, September 1997.

[10] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, and G. Swallow, “RSVP-TE: Extensions to RSVP for LSP Tunnels,” *RFC 3209*, December 2001.

[11] L. Berger, “Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReSerVation Protocol-Traffic Engineering (RSPV-TE) Extensions,” *RFC 3473*, January 2003.

[12] D. Papadimitriou and D. Verchere, “GMPLS User-Network Interface in Support of End-to-End Rerouting,” *IEEE Communication Magazine*, pp. 35 – 43, July 2005.

[13] D. Papadimitriou, B. Rousseau, W. Krber, S. Brockmann, and D. Verchere, “The Private User Network Interface - a GMPLS-compliant signaling interface for overlay-based multilayer networks,” *Journal of Optical Networking*, pp. 119 – 132, March 2004.

[14] C. Margaria, G. Juillot, and A. Autenrieth, “Performance Evaluation of a GMPLS Prototype,” *Broadband Communications and Distributed Systems*, pp. 211 – 222.

[15] S. D. Maesschalck, M. Pickavet, D. Colle, and P. Demeester, “Multi-layer Traffic Grooming in networks with an IP/MPLS Layer on top of a meshed Optical Layer,” *GLOBECOM*, pp. 2750 – 2754, 2003.