

Emulated MPLS-ASON/GMPLS Inter-connection test-bed

Gerson Rodríguez de los Santos¹ and Ricardo Romeral¹

Luis Velasco², Fernando Agraz² and Salvatore Spadaro²

¹*Dpt. of Telematics, Universidad Carlos III de Madrid (UC3M), Madrid, Spain
E-mail: {gerson, rromeral}@it.uc3m.es*

²*Advanced Broadband Communications Center (CCABA)
Universitat Politècnica de Catalunya (UPC), C/ Jordi Girona, 31. 08034 Barcelona Spain
E-mail: lvelasco@ac.upc.edu, {agraz, spadaro}@tsc.upc.edu*

Abstract

In this paper we show how use different emulation technologies to establish a test-bed. The presented test-bed emulates a multi-domain network, two IP/MPLS islands interconnected through an ASON/GMPLS network. The ASON/GMPLS grooming-capable CARISMA network test-bed and Dynagen/Dynamips emulation technology are presented as integral part of the test-bed. This test-bed allows do research in multi-domain multi-layer protection mechanisms in the future Optical Internet.

Index Terms— Test-bed, MPLS-GMPLS interworking

Introduction

The introduction of intelligence in Automatically Switched Optical Networks (ASON) [1] using a Generalized Multiprotocol Label Switching (GMPLS) control plane [14] allows to setup, configure, and release optical connections, in a fast and dynamic way.

In a close future the well-known IP/MPLS network must coexist with the new GMPLS-controlled optical networks and the interaction between them will be inevitable.

One of the future network scenarios will be one or more IP/MPLS networks interconnected through an ASON/GMPLS network [3]. In this scenario interaction between the MPLS and ASON/GMPLS will be needed, especially in signaling protocol to allow set-up end-to-end LSPs with the desired QoS. One of the most common ways to provide QoS in circuit-based networks is by providing LSP protection.

For these reasons, the IETF Common Control and Measurement Plane Working Group (CCAMP) [4] has the standardization of the MPLS-GMPLS interaction as one of their scopes.

To advance in future Optical Core Network research is essential to get the appropriate simulation tools, since research in real optical networks requires high investments. There are two main ways to do this, to simulate the network and to emulate it.

In this paper we show an IP/MPLS ASON/GMPLS network test-bed. This test-bed is based on an emulated IP/MPLS test-bed which uses two different tools, dynagen and dynamips, and the CARISMA network test-bed, an emulated ASON/GMPLS network. Both test-beds have been interconnected via Internet GRE Tunnels.

Using this test-bed research community could advance in new signaling procedures to improve future optical network survivability in a multi-domain and multi-layer scenario.

The paper is structured as follows. First, we present the CARISMA network, followed by the Dynagen/Dynamips emulation environment. The interconnected test-bed emulating a multi-domain MPLS ASON/GMPLS network is showed, previously to the conclusions section.

The CARISMA Network Test-Bed

The CARISMA network test-bed has been implemented to be used as a multi-domain field-trial for the integration and evaluation of the ASON/GMPLS technologies.

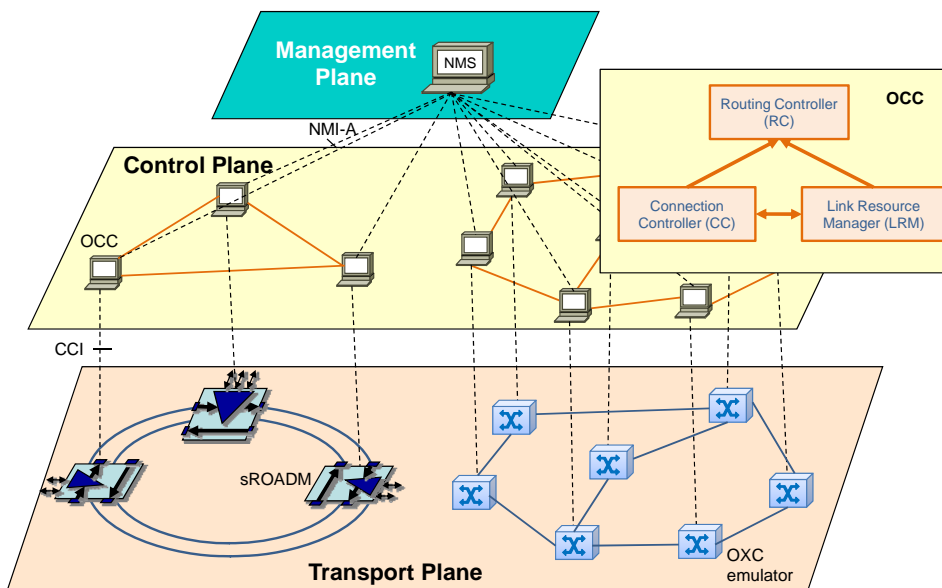


Fig. 1 The CARISMA network test-bed

Fig. 1 presents the architecture of the CARISMA network test-bed. It contains the following three functional planes: the transport plane, responsible for traffic transport and switching; the control plane, responsible for connection and resource management. It can be either associated with (in-fiber) or separated from (out-of-fiber) the managed transport network; and the management plane, responsible for management of the whole system (including transport and control planes). It triggers commands to the control plane to set-up and tear-down soft-permanent connections. At the management plane of

the ASON/GMPLS CARISMA network test-bed, the Network Management System (NMS) was implemented as a web-based application, easing network management through the Internet.

At the transport plane, two alternative nodes can be used: the physical node and the emulated node. At the control plane, every optical node connection controller (OCC) contains three modules which communicate among them.

The Transport Plane

The CARISMA transport plane uses both physical and emulated optical nodes. Using the physical nodes, called semi-reconfigurable optical add/drop multiplexers (OADM) (sROADM), a unidirectional ring can be build. However, in order to perform tests over more complex networks, including mesh networks, optical cross-connects (OXC) node emulators have been developed. Both, the sROADM and the OXC emulators provide the same Connection Control Interface (CCI) making the concrete configuration of the transport plane transparent to the control plane.

This optical node allows dropping two wavelengths out of the bundle of dense wavelength division multiplexing (DWDM)-multiplexed signals and adding two new wavelengths to the DWDM- multiplexed bundle. The monitoring device extracts a small part of the incoming optical power, transforms the sample into a digital value by means of an A/D converter, and stores the converted value in a register. The monitoring sweep time is 10 μ s.

Two demultiplexer/multiplexer modules have been used in order to extract/insert 8 contiguous DWDM 100GHz separated wavelengths. The mux/demux module has insertion losses lower than 3.5 dB. All remaining channels are reflected onto the express port which can be used to connect a full Reconfigurable OADM module.

The sROADM node is equipped with a 4x2 optical switch fabric, in order to select the wavelengths to drop. The 4x2 optical switch has been built using the simplest 2x2 and 2x1 optical switches. The switching device has a very fast response (t_{switch}) time below 1ms and has insertion losses lower than 0.9 dB. The sROADM uses a 100 Mbit/s Ethernet signal over the optical supervisory channel (OSC), to convey the control plane information when the *in-fiber out-of-band* option (described in the next section) is used. Moreover, specific mux/demux components are equipped in order to attach/detach the OSC 1310nm optical signal to/from the DWDM-bundle.

As introduced previously, OXC emulators are needed to build complex topologies. Moreover, OXC emulators allow for testing new optical nodes architectures and network solutions. Regarding this, we use OXC emulators to measure times related to protection/restoration. Besides the CCI interface, the OXC emulator implements an additional interface to receive failure information related with the equipped ports.

The OXC emulator provides configurable delays when implementing SNCs. Upon reception of a command through the CCI, the request is stored in a *first-input first-output* request queue. A connection processor is continuously retrieving commands from the queue and executes the command introducing the specified delay, emulating the switching time (t_{switch}) of a physical device.

The Control Plane

The CARISMA GMPLS control plane uses the RSVP-TE protocol for signaling, the OSPF-TE protocol for routing, and the LMP protocol for control channel management and link property correlation. The OCCs have been implemented using Linux-based routers. Each pair of OCCs communicates through a single IP control channel implemented with full duplex Fast Ethernet links. Finally, each OCC communicates with the local OXC through the CCI.

The OCC contains three modules which communicate among them, as shown in Fig. 1: the Link Resource Manager (LRM), the Routing Controller (RC), and the Connection Controller (CC).

The LRM module (Fig. 2) is responsible for the management of the resources available at the optical node. Note however that, in the extended version of the LRM available in the CARISMA control plane, it manages also control plane resources and implements additional functionalities such as control channel management, resource discovery, control plane recovery, etc. The transport plane resources state is stored in the MIB, and the OXC Manager module synchronizes the state of those resources with the optical node through the CCI. The same interface is used by the node to notify alarms using Simple Network Management Protocol (SNMP) traps [5]. Whenever the node used, physical or emulated, the view of that node at the control plane is the same. A node contains a set of ports which are connected to TE-links in a one-to-one relationship. Every port contains a set of connection points (CP) [6] and every TE-link contains a set of data-links. CPs and data-links are associated and represent every wavelength in the DWDM-bundle. Note that CPs and data-links are unidirectional, while TE-links and ports can be bidirectional when encompassing different input and output data-links/CPs.

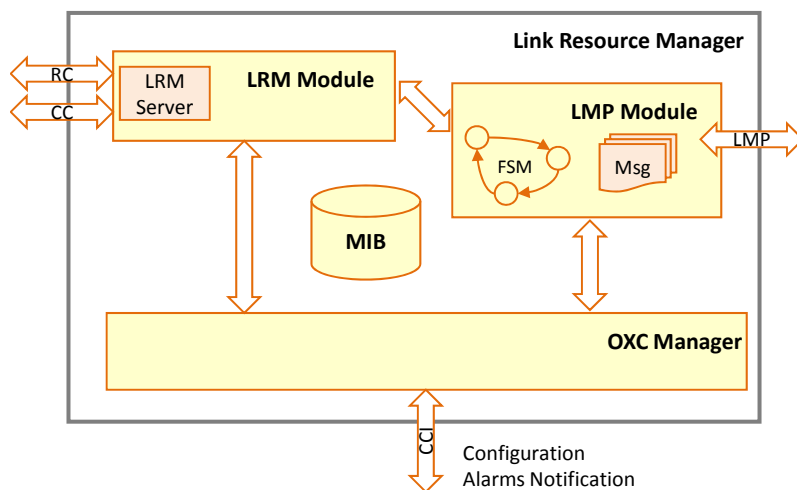


Fig. 2 Link Resource Manager

Moreover, the LRM contains the LMP module, which implements the Link Management Protocol (LMP)[7], and the LRM server module which implements

several communication interfaces with the RC and CC modules belonging to the same OCC.

The RC (Fig. 3) is responsible for route computation. It implements several routing algorithms to compute control plane routes. In this regard, it is worth noting that the CARISMA network test-bed uses differentiate addressing spaces at the control plane and at the transport plane. In fact, the *quagga* OSPF module [8] implements the OSPF protocol which floods links state information related with the control plane IP network. Nonetheless, the RC floods the state of the local outgoing data-links, using OSPF-TE Opaque Link State Advertisements (OLSA) [9], [10] to its control plane neighbor OCCs. The information in the OLSAs is related with the transport plane and it is stored in the TE database. OLSA flooding is performed every time a data-link is used by a LSP or is released. The RC module implements communication interfaces with the CC and with the LRM. The CC requires route computation between two end nodes, whereas the LRM notifies the RC about the whether a local resource has been used or released and about failures. That state changes imply OLSA flooding.

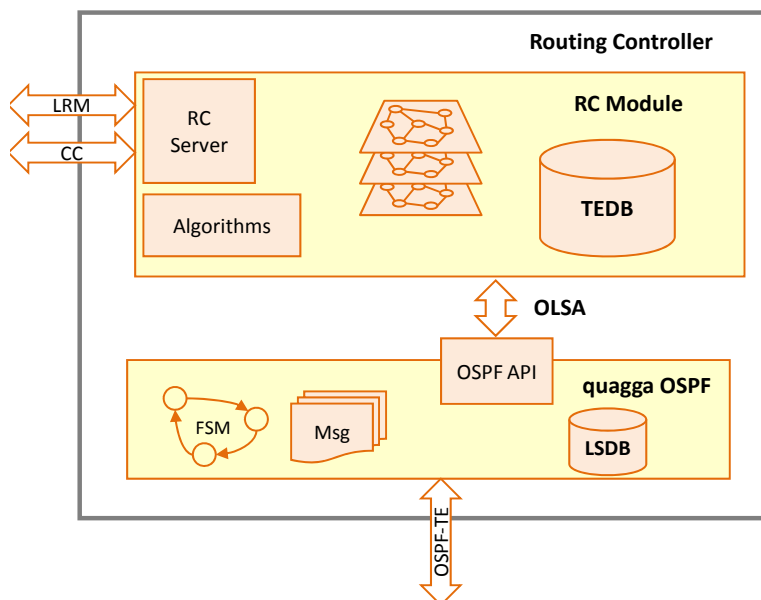


Fig. 3 Routing Controller

Finally, the CC (Fig. 4) is responsible for LSP set-up and tear-down. The CC module includes the RSVP module which implements the RSVP-TE protocol [11], [12]; **Error! No se encuentra el origen de la referencia., ¡Error! No se encuentra el origen de la referencia.** The CC contains the path state block (PSB) database which stores every LSP already establish using resources at the local optical node. The NMS communicates with the CC through the NMI-A interface to request set-up or tear-down connections. Upon the reception of a set-up command between two end nodes, the CC asks the RC for a route at the transport plane. Moreover, every CC in the route of a setting-up LSP must ask the LRM about the availability of the local resources and request it to allocate those resources.

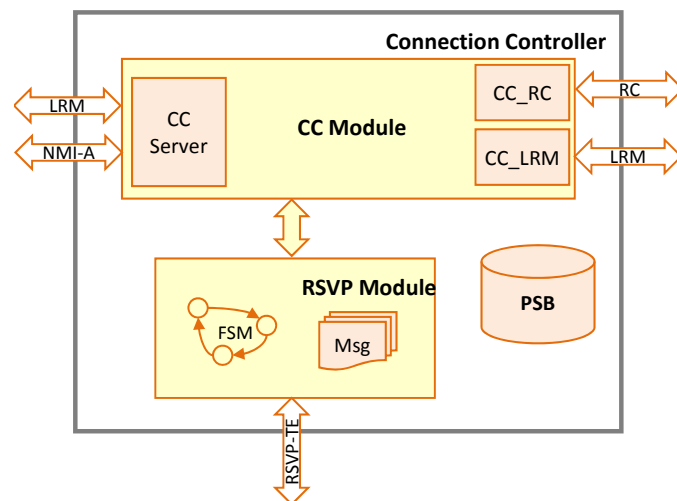


Fig. 4 Connection Controller

Dynamips/Dynagen test-bed

Dynamips is a Cisco Emulator created by Christophe Filliot [Ref]. At first, the project consisted on emulating a Cisco 7200 router but, as the project progressed, it other families such as the 3600, 3700 and 2600 series. It provides support not only for Ethernet interfaces, but also for ATM (Asynchronous Transfer Mode), Serial and PoS (Packet over SONET) interfaces.

Dynamips emulates the underlying hardware of the router in order to run a Cisco IOS image. The user can specify the interface modules the emulated router has and the slot in which these modules are placed. The configuration of the router is passed through the command line. Dynamips also has a server mode in which another process can connect and create new router instances. It also allows changing the router hardware configuration during the emulation. The routers' interfaces can be mapped to virtual interfaces (in the case of Ethernet interfaces), sockets (in which case frames are encapsulated in UDP or TCP datagrams) and physical interfaces. This last feature means that we can map a router interface to a real Ethernet card, which can give access to a real network to our virtual routers in order to connect to other virtual networks or even to real routers. Another interesting feature is that we can choose the amount of RAM and NVRAM (Non Volatile RAM) of the virtual routers as well as the way in which the operating system of the computer holding the emulation will manage memory, permitting us to map the RAM of the routers to a swap file instead of the RAM of our computer.

This server mode is used by another application, Dynagen [Ref], to automatise the creation of emulated network scenarios. Dynagen is a front-end for Dynamips. Dynagen is able to read network description files in a simple syntax. These files describe the routers present in the network, the interfaces they have and how they are connected to other routers by either virtual or physical interfaces. It also provides a console prompt to

give basic commands to dynamips, such as starting and stopping a router or even disconnecting an interface from the router. Furthermore, we can save the configuration of our routers to the text file where our scenario is described so we can share it with other people.

The emulated IP/MPLS test-bed can be seen in Fig. 3.

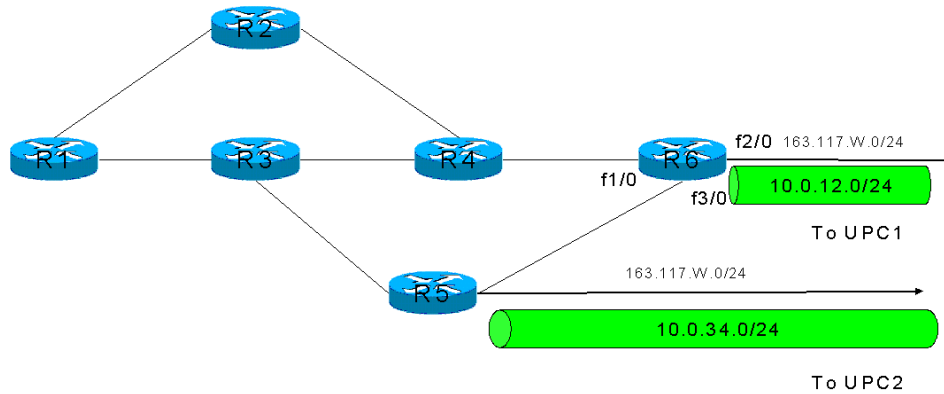


Fig. 3 The IP/MPLS emulated network

It consists of six virtual Cisco 7200 routers emulated with the Dynamips Cisco emulator. All routers are connected with virtual Fast Ethernet interfaces except for the link R3-R4, which is a PoS link. The links between the nodes were chosen to build a trap topology. Routers R5 and R6 are directly connected to the Internet by mapping two of their interfaces to one of the wired network cards of the server where the emulation is held. Each one of these two interfaces is given a public IP address.

MPLS-ASON/GMPLS Network interoperation

As we can see in Fig. 4, the two routers accessing the Internet have established a GRE tunnel with the network at the UPC. GRE tunnels allow us to encapsulate MPLS frames through the Internet, giving continuity to the MPLS domain we have defined. This way, we can connect several virtual networks hosted in different servers of the Internet. This allows us to simulate large networks in a distributed way. Should several links fail and our network be splitted into two parts, we could connect to the other half via the UPC network.

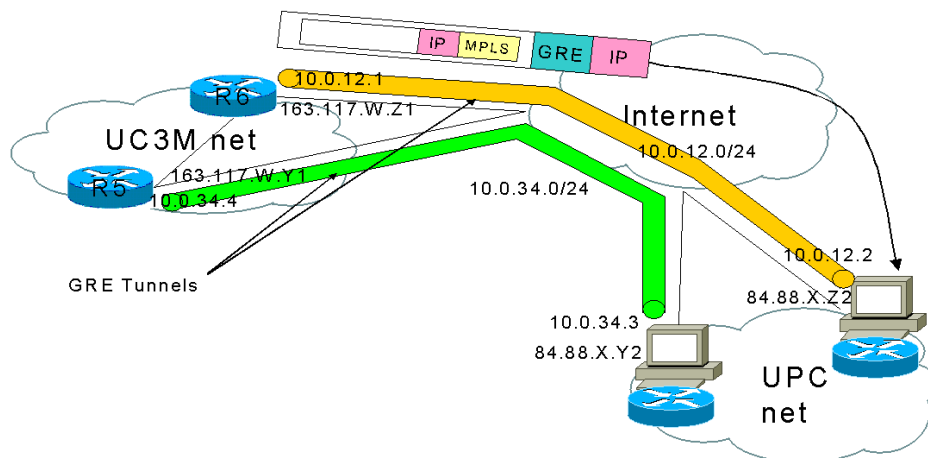


Fig.4 MPLS ASON/GMPLS interconnection test-bed via Internet

Over this test-bed we can test interoperation between different G/MPLS implementations and different multi-domain G/MPLS protection schemes and multi-layer protection schemes as well.

First end-to-end multi-domain LSPs (non-protected) have been established already in the test-bed. After that a whole world of research opportunities is opened.

Conclusions and Further Work

In this paper we have presented two emulated networks, one of them an ASON/GMPLS network based on the CARISMA network test-bed and the other one an IP/MPLS Network, based in Dynagen/Dynamips emulated environment. The resulted test-bed is a distributed emulated multi-layer multi-domain network. Over this network different scenarios can be studied in different aspects. Our intention is to use it to research in future protection schemes for the future Optical Internet.

The test-bed allows studying different inter-domain survivability schemes, doing possible to compare along them and select the best one in each scenario and application.

Using the showed technology test-beds can be larger and simulate a thousand nodes Network, given the opportunity to probe the different signaling procedures in a *real-like* Network.

Our main goal is to use the test-bed to probe the proposed mechanism in [3] and resolve the signaling interoperation problems between a MPLS Network and an Optical GMPLS Network.

Acknowledgment

The work described in this paper was carried out with the support of the BONE-project ("Building the Future Optical Network in Europe"), a Network of Excellence funded by the European Commission through the 7th ICT-Framework Programme.

References

- [1] ITU-T G.8080, "Architecture for the automatically switched optical network (ASON)," (2001).
- [2] E. Mannie, "Generalized Multi-Protocol Label Switching (GMPLS) Architecture," RFC-3945 (2004).
- [3] "On the design of MPLS-ASON/GMPLS Interconnection Mechanisms". Luis Velasco, Ricardo Romeral, Fernando Agraz, Salvatore Spadaro, Jaume Comellas, Gabriel Junyent, and David Larrabeiti. VII Workshop in G/MPLS Networks. Vilanova i la Geltrú, Cataluña, Spain, 11-12 Marzo 2008.
- [4] CCAMP, <http://www.ietf.org/html.charters/ccamp-charter.html>
- [5] D. Harrington, R. Presuhn, and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks," RFC 3411, (IETF, 2002).
- [6] "Generic Functional Architecture of Transport Networks," G.805 (ITU-T, 2000).
- [7] J. Lang, "Link Management Protocol (LMP)," RFC 4204 (IETF, 2005).
- [8] GNU Quagga Routing Software. <http://www.quagga.net>
- [9] D. Katz, K. Kompella, and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2," RFC 3630 (IETF, 2003).
- [10] K. Kompella, Y. Rekhter, "OSPF Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)," RFC 4203 (IETF, 2005).
- [11] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels," RFC 3209, (IETF, 2001).
- [12] L. Berger, "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions," RFC 3473 (IETF, 2003).
- [13] ITU-T G.8080, "Architecture for the automatically switched optical network (ASON)," (2001).
- [14] E. Mannie, "Generalized Multi-Protocol Label Switching (GMPLS) Architecture," RFC-3945 (2004).